Secure Patient Portal

Muskaan Verma
B.Tech (CSE)
MIET, MEERUT

Vrinda Singhal
B.Tech (CSE)
MIET, MEERUT

Tanya Verma
B.Tech (CSE)
MIET, MEERUT

Vishal Jayaswal
Assistant Professor
Department of Computer Science & Engineering
MIET, MEERUT

ABSTRACT

Secure Patient portal is designed in such a way that in these uncertain times of Covid-19 pandemic, patients can book their appointments online as per the convenience of them as well as doctors also. Instead of waiting for longer hours at the hospital they can schedule their appointments and reach on time .It will prevent them from unwanted contact with other patients. It will also prevent the patient's medical history, doctor's appointment schedules and other sensitive data from SQL injection attacks. Through this, it will be easy for doctors to manage the appointment slots online. Also there are many advantages for patients as they can see their current visit information, appointments of doctors and can communicate securely with their healthcare providers. Through this system it is easy to manage availability of various doctors in terms of dates and timings as per the requirements. It is a very secure portal as it prevents the portal's sensitive information from OWASP's SQL injection attack**.**

Keywords

SQL Injection, Anytime, Anywhere, Secure

1. INTRODUCTION

We are making this secure patient portal so that both patients and doctors will benefit from this. This system allows patients to interact with their doctors and healthcare providers. But this system can't replace doctors in- office visit. This system will definitely help doctors in their work and also patients can book their appointments with doctors and can see their medical progress. It will help both doctors and patients to book their slots online .Patients can see which slots of their doctors are free and according to it they can reserve their slots. We are creating a secure patient portal with one of the most important security measures that is ideal for a health care system. Patient portal is a website, we also stick to the OWASP's SQL Injection security measure due to which the sensitive data of the portal will be secured from any unwanted attack. No outsider can breach into our system and misuse the information.

2. LITERATURE SURVEY

The existing system is a manual system in which the patients have to go to the doctor's clinic and wait for a longer time for their turn to come **S**ecure Patient portal is designed in such a way that in these uncertain times of Covid-19 pandemic, patients can book their appointments online as per the convenience of them as well as doctors also. As instead of waiting for longer hours at the hospital they can schedule their appointments and reach on time .It will prevent them from unwanted contact with other patients. It will also prevent the patient's medical history, doctor's appointment schedules and other sensitive data from SQL injection attacks .Through this, it will be easy for doctors to manage the appointment slots online. Also there are many advantages for patients as they can see their current visit information, appointments of doctors and can communicate securely with their healthcare providers. Through this system it is easy to manage availability of various doctors in terms of dates and timings as per the requirements. It is a very secure portal as it prevents the portal's sensitive information from OWASP's SQL injection attack. Also there is no feedback facility in the manual system so there will be no analysis on positive and negative responses and also does not tells whether the doctor and the medical staff is good and taking good care of patients or not. Also, if there are any changes required for improvement. Therefore it does not lead to any improvement that is required in the system .This all problems can be solved by our SECURE PATIENT PORTAL since it has an availability of feedback for the patients.

2.1 Framework and Approach

It is a very effective and secure portal for patients .It is changing the way the patients think about the health care systems. Through these portal patients can schedule appointments, view their health records, access their medical history and access medication and discharge instructions. The patient's data which is integrated with the labs is extracted from the hospital's database. The theme of this portal is "Anytime, anywhere". Through which patients are able to access the Health information anytime anywhere without worrying about their data's security. In this system the admin create a new patient. For the first time when the patient login, they need to reset their password, which is defaulted by the system for the security of our health information.

We are using SQL Injection of "Open Web Applications Security Project" (OWASP) to ensure the security of our web application. We are using this to secure our application from the SQL attacks .SQL injection belongs to the category of code injection. SQL injection vulnerabilities are considered as one of the most serious threat to the web applications. Most of our web applications are database driven that means a database is used to store the valuable information. Due to this reason, the applications will be processing or executing various types of SQL queries. If our web application is vulnerable to SQL injection, it may allow the invader to gain complete access to the database containing the patient's personal health information. Due to which they can insert, update, or delete the valuable information. User input is an important part of this type of application, because of the filter conditions performed by these queries. Input validation can be used to prevent the SQL injection .In this research we are going to use LINQ to SQL for all the database operations. Traditional SQL queries concatenated with user input has been a significant risk of SQL injection.

In LINQ to SQL the SQL injection is avoided by using SQL parameter in queries. The input entered by the user is converted into parameter values. This approach is used to avoid the malicious commands being entered by the customer. Broken authentication and session management attacks are anonymous attacks that are generated to access the confidential data i.e. retrieve passwords, user-id and other information. The attackers begin the attack by relying on the information they have collected from the users of the same platform.

3. BACKGROUND OF SQL INJECTION ATTACKS

Through data-driven web applications SQL injection attack is the one of the most used types of OSWAP's attack taken by cyber attackers. In this approach we use SQL commands such as Delete, Select, Update and Insert etc. These commands help these harmful attackers to design the actual SQL code again efficiently. We execute unsafe code into the web applications through these SQL commands. When these harmful attackers reach their goal then: private information is easier for them to access, execute and modify secured data. Also the whole application can be failed by them suddenly. Unofficial accesses can lead to loss of privacy of the database administrator and their role. By this, attackers are highly paid through SQL injection attacks because they focus on stolen bank accounts, credit card numbers and so on. Security issues of this type are more affected on web applications, and can be handled by the authentication of users. Many forms of SQL injection attacks are there. Commonly used attacks take the benefits which are as follows: The "i" parameters are incorrectly passed, type handling is incorrect, SQL statements contain many errors for e.g. ('OR) 1 = -- '). There are many different types of SQL injection attacks are there: i.e. tautologies, illegal/logically incorrect queries, Piggy-backed queries, Stored Procedures, UNION query Blind SQL, Timing Attack and Alternate Encoding and so on. As the attackers manipulate the behavior of original SQL queries so it is not easy to conquer these attacks.

3.1 Workflow of Proposed Plan

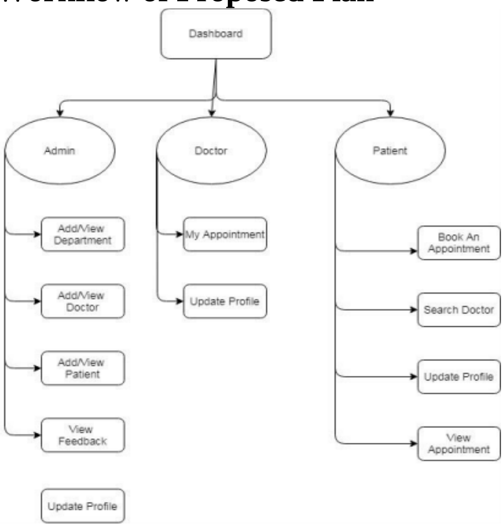

Fig 1: Workflow of Proposed Plan

3.2 Proposed algorithm used

FLOWCHART

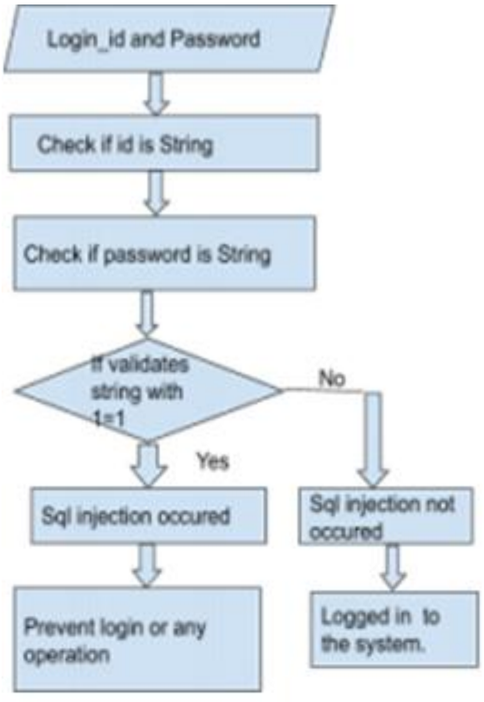
Login_id and Password
Check if id is String
Check if password is String
If validates string with 1=1
No
Yes
Sql injection occured
Sql injection not occured
Prevent login or any operation
Logged in to the system.

Fig 2: Flowchart for sql injection

4. EXPERIMENTAL RESULT ANALYSIS

Total attacks to the total detection is called detection rate. We cannot find the intrusion in actual observation. For the observation and removal of SQL intrusion. Dynamic analysis is not an appropriate method. We can find the exposure by using dynamic analysis methods in web applications. Java and SQL Languages are used to do the SQL injection free analysis. This technique is common in any web applications, language independent and can be executed in any programming language. For the observation of SQL injection attacks. Source of the program plays an important part in this approach. The results calculated are positive when we calculate this algorithm in real time during an initial time. It needs further investigation with vulnerable web applications for this web application and it is free from SQLIAs.

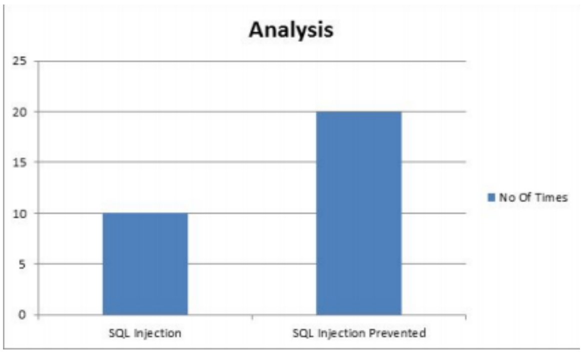

Fig 3: Analysis Chart

It denotes that if we enter the data 30 times into the portal then out of 30, 10 times SQL injection occurs whereas 20 times SQL injection is prevented.

5. CONCLUSION

For easy observation in opposition to Structured query language pop intrusion, this algorithm plays an important role. For ensuring this algorithm performance and number of times quality, implementation for Structured query language pop intrusion is an important task. To determine against SQL injection attacks this algorithm provides much quick and with accomplished solution .With help of different detection methods and the suggested method it cannot only be executed on web applications but also can be executed on any applications which directs towards databases this is examined by paper work. To combine SQL parser is the main motive of future research. One difficult method is to create a parser to detect critical vulnerabilities in this algorithm. In three-tier for solidify these implementations WWW facility dynamic checking compiler can be used and for protecting from Structured query language intrusion. To achieve effectiveness and regulation both these approaches were quite practical and efficient.

6. REFERENCES

[1] The Open Web Application Security Project(OWASP), Available:hnps:l/www.owasp.org index.php/Top 10 2013-Top 10.

[2] https://owasp.org/wwwcommunity/attacks/SQL_Injection

[3] Healthcare IT News,Available:hnp:llwww.healthcareitnews.com/directory/patient-portals.

[4] https://www.acunetix.com/websitesecurity/sql-injection/

[5] Microsoft Azure, Available: http://azure.microsoft.com/en-us/.

[6] https://ieeexplore.ieee.org/document/891201

[7] https://ieeexplore.ieee.org/document/7280212

[8] https://projectideas.co.in/efficient-doctor-patient-portal-dotnet/

[9] https://sucuri.net/guides/owasp-top-10-security-vulnerabilities-2020/

[10] https://docs.google.com/document/u/1/d/1N1xoedIACfvM_ScbHVRKpxstNvPLgTogHfINo Yx6Oy0/ mobilebasic.