

**School of Computer Science & Engineering**

**Project Report**

**Presented To -** Bhulakshmi Bonthu

**Course Code** : CSE3001 (J component)

**Slot – E2**

**Course Name : Web Security**

**Project Title - Detection and Prevention of SQL and Cross Site Scripting Attack**

**School - SCOPE**

**Team members –**

Yashraj Agarwal (18BCI0183)

Abhinav Gupta (18BCI0204)

Divyansh ()

**Chapter 1:**

* 1. **ABSTRACT:**

Organized Query Language (SQL) infusion and cross-website scripting stay a significant danger to information driven web applications. Occurrences where programmers get unlimited admittance to back-end information base of web applications to take, alter, and wreck private information are expanding. Subsequently, measures must be set up to abridge the developing dangers of SQL infusion and XSS assaults. This investigation presents a procedure for recognizing and forestalling these dangers utilizing Knuth-Morris-Pratt (KMP) string coordinating calculation. The calculation was utilized to coordinate client's info string with the put away example of the infusion string so as to distinguish any malevolent code. The usage was completed utilizing PHP scripting language and Apache XAMPP Server. The security level of the strategy was estimated utilizing diverse experiments of SQL infusion, cross-site scripting (XSS), and encoded infusion assaults. Results acquired uncovered that the proposed method had the option to effectively distinguish and forestall the assaults, log the assault section in the information base, block the framework utilizing its macintosh address, and furthermore produce an admonition message. Along these lines, the proposed procedure end up being more viable in identifying and forestalling SQL infusion and XSS assaults

**1.2Acknowledgement:**

We are grateful to VIT University for encouraging and giving us the opportunity to perform this project. We would like to thank our Project Guide, Professor Jayakumar K. for her guidance and support through every step of this project. Last but not the least, I am thankful to all family and friends for their help. The completion of this project is attributed to everyone’s combined effort.

**1.3 Introduction :**

Web is quick turning into a family innovation with 4.39 billion clients in January 2019 contrasted with 3.48 billion clients in January 2018 . This demonstrated that more than 1,000,000 new clients got associated every day. This development rate is being encouraged by information driven web applications and administrations which empower clients to execute their online exercises easily. Most present day associations and people vigorously depend on these web applications to contact their various clients. Clients' information sources through web applications are utilized to inquiry back end information bases in order to give the required data. This pattern has consequently opened up web applications and administrations to assaults by programmers. Additionally, the notoriety of web application in long range interpersonal communication, monetary exchange, and medical issues are expanding quickly; accordingly, programming weaknesses are turning out to be basic issues, and in this manner, web security has now become a significant concern . The weaknesses are generally application layer weaknesses, for example, area name worker assaults, Inline Frame defects, far off record incorporation, web validation blemishes, far off code execution, XSS, and SQL infusion . An overview completed by Open Web Application Security Project (OWASP) recognized top 10 weaknesses as at June 2019 to be infusion blemishes, broken validation and meeting the executives, touchy information introduction, XML outside element, broken admittance control, security misconfiguration, XSS, uncertain deserialization, utilizing segments with known weaknesses, lacking logging, and observing. In any case, among these types of assaults, XSS and SQL infusion have been distinguished as the most hazardous . The WordPress Security Learning Center likewise presents that in the event that SQL infusion and XSS weaknesses could be dealt with in a code, at that point 65% weaknesses has been wiped out. Since web applications use information provided by clients in SQL inquiries, programmers can control these information and supplement SQL meta-characters into the info fields in order to get to, change, or erase the substance of the data set. For example, the WHERE proviso in the SQL question SELECT\*FROM clients WHERE secret key = 1234 could be controlled when programmers flexibly inputs like 'anything' OR '1' = '1'; #. The WHERE provision currently contains two conditions isolated with the coherent administrator OR. The primary condition probably won't be TRUE, yet the subsequent condition must be TRUE since 1 is consistently approaches 1, and the intelligent administrator "OR" returns TRUE assuming either or both of the conditions are TRUE. Henceforth, the programmer obtains entrance without a need to know the secret key. At times, wrong information esteems can likewise be provided deliberately so mistake messages that will assist the aggressors with understanding the data set composition will be uncovered. In this manner, SQL infusion is a genuine danger for web application clients.

**1.4 Organization of the report –**

**Chapter 4:**

**Conclusion –**

**4.1 Summary –**

In this report, we contemplated the example that follows both SQL infusion and XSS assaults, and separated it into segments so as to perform different end calculations to forestall the assault. During the filtration, we checked for specific examples, and on the off chance that it exists, prompt end happens and the assault is forestalled. Since Cookies assume a significant part during an assault the majority of the time, we set forth two techniques in particular encryption and hashing to forestall the equivalent. So as to check the weakness of different sites, we endeavored to abuse XSS through the Burp Suite Software.

As the third procedure, we made a site and actualized certain highlights to ensure burglary and some other kind of unapproved access.

A comparison of the proposed technique with existing techniques revealed that the proposed technique is more efficient because it is not limited to a particular form of attack, and it can handle different forms of SQL injection and XSS attacks.