**An Approach to Prevent Authentication & Authorization Exploit by SQL Injection Using Hashing & Encryption (SQLHE)**

**Abstract--** SQL injection is hacking technique where the term injection refers to insertion of suspicious or malicious query inside login phase. Before Login SQL Injection Attack (BL-SQLIA) is where the hacker exploits the authentication & gain illegal access into the system. After Login SQL Injection Attack (AL-SQLIA) is where the hacker exploits the authorization & can access the contents using after login sql injection attacks which includes union based query attacks, stored procedure attack & Havij tool attacks etc.

It is known that security provided in existing system at before login phase is efficient but there are chances that legal user can access the contents of other user once it is inside the system.

Therefore we implement our proposed scheme called SQLHE, SQL injection prevention by hashing & encryption techniques to handle the SQLIA & prevent them. Proposed implemented system fails all kinds of online decrypter along with after login attack tools like Havij tool attacks, ExploitUnion tool attacks etc. However proposed technique is completely new to user but it can be applied to Databases like ORACLE, MSSQL, and MYSQL etc.

**Keywords**

SQL injection, SQL injection attacks, BL-SQLIA, AL-SQLIA, SQLHE, legal user, hashing, encryption etc.

**I. INTRODUCTION**

Information is the most important business asset in today’s environment and achieving an appropriate level of Information Security is must to be achieved. SQL-Injection Attacks (SQLIA’s) are one of the topmost threats for web application security. For example financial fraud, theft confidential data, deface website, sabotage, espionage and cyber terrorism.To implement security guidelines inside or outside the database it is recommended to access the sensitive databases should be monitored. In this project, two techniques HASHING and ENCRYPTION are proposed for SQL injection prevention. Initially hashing of username and password is to be done then we perform Ex-OR function on them for “Before Login” security. After that we apply ENCRYPTION to Database for providing security to backend and prevent “After Login” attack at server side. Most of us either use Web applications on a daily basis, as part of our vocation or in order to access our e-mail, book a holiday, purchase a product from an online store, view a news item of interest, and so forth. One thing that Web applications have in common, regardless of the language in which they were written, is that they are interactive and, more often than not, are database-driven. Database-driven Web applications are very common in today’s Web-enabled society. They normally consist of a back-end database with Web pages that contain server-side script written in a programming language that is capable of extracting specific information from a database depending on various dynamic interactions with the user. One of the most common applications for a database-driven Web application is an ecommerce

application. While, on the other side, the organization increases the use of office automation

software & services, that helps them to maintain the confidential information with less efforts. Therefore, in this scenario it is not wrong to say that Information will be the single most important business asset today and achieving a high level of information security can be viewed as imperative in order to maintain a competitive edge. SQL Injection Attacks (SQLIA’s) are one of the most severe threats to web application security. They are frequently employed by malicious users for a variety of reasons like financial fraud, theft of confidential data, website

defacement, sabotage, etc. The number of SQLIA’s reported in the past few years has been showing a steadily increasing trend and so is the scale of the attacks. It is, therefore, of paramount importance to prevent such types of attacks, and SQLIA prevention has become one of the most active topics of research in the industry and academia. There has been significant progress in the field and a number of models have been proposed and developed to counter SQLIA’s, but none have been able to guarantee an absolute level of security in web applications, mainly due to the diversity and scope of SQLIA’s. This paper is divided into following section, In Section I introduction, in section II basics of SQL injection, in section III various techniques of SQL injection attack, in section IV effects of SQL injection, in section V our proposed approach, in section VI implementation details and in section VII conclusion and future work is to be present.