ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 6, June 2016

SQL Injection (SQLi)

Sejal Farde, Sailee Chaudhari

Department of MCA, Mumbai University,

Institute of Management & Computer Studies, Thane (W)

***Abstract* –**
SQL injection is a code injection technique which is use to attack data-driven application, in which malicious SQL statements are inserted into an entry field for execution. SQL-Injection must exploit a security vulnerability in an application 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 & 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 injection is the type of attack in which the attacker adds SQL code to the web form input box to gain access or make changes to the data. SQL injection vulnerability allows attacker to flow commands directly to a web application's underlying database & destroy functionality & confidentiality. Researches have proposed different tools to detect & prevent these vulnerabilities. In this paper, we present all SQL injection attack types & also tools which detect or prevent this attacks. SQL injection attacks allow attackers to spoof identity, cause repudiation issues, allow complete disclosure of all data on the system, destroy data or make it otherwise unavailable, & become administrators of the database server.

***Keywords* – SQL Injection Attacks, Detection Tools, Prevention.**

I – INTRODUCTION

SQL Injection firstly appeared in 1998's Phrack Magazine which was public. Since web applications have become the most important conversation channels between every service providers & customers, script masters & hackers target victims for fun, or for personal advantages or for commercial purpose. The increased frequency & complexity of web applications has raised huge awareness among web application administrators to effectively protect their web based applications & implement strict security.

Various web applications such as net banking, online shopping, certain employee portals are all available to their respective customers &all attackers due to their 24X7 online nature on internet. Attacks such as session hijacking, SQL Injection, etc. are aimed at vulnerabilities in web application itself. "Firewalls" (one of the specialized tool) is use to serve the purpose of strong security in our web application.

The web application security has become the most important part in day-to-day lives & majority of the firms concentrate on this aspect. Data security involves several threats & hence every organization is involved in high security of their web based applications.

Structured Query Language (SQL) injection is a technique to attack the database. Here the exploitation of security is done in a way wherein the attacker adds the SQL code to a web form input box and by doing so the attacker can gain access to database or can also make changes to the data.

SQLiis declared as one of the top 10 Web Application vulnerability of the years 2007 & 2010 by OWASP (Open Web Application Security Project).

II – TYPES OF SQL INJECTION

SQLi can cause serious problems in various ways. In certain situation SQLi is use to execute commands on OS, by allowing attacker to cause more damage inside the network behind the firewall. In SQL injection levering, destroyer can access, modify & delete within dB by using bypass authentication mechanism.

There are **3 main categories** of SQL Injection

1. In-Band SQLi
   1.1. Error based SQLi
   1.2. Union based SQLi
2. Inferential SQLi
   2.1 Content based SQLi
   2.2 Time based SQLi
3. Out-of-Band SQLi

1. **In-Band SQL Injection**–Also known as "**Classis SQLi**". It is the most common & easy to exploit SQL injection attack. This attack occurs when destroyer gets access to the same communication channel as that of sender/receiver, & then he launches the attack & gathers results.

The **2 sub classes** of this category type are

Error based SQL Injection - It depends on the error messages that are displayed by dB server for obtaining the information about the dB structure. Sometimes, this type itself is much more enough to enumerate the whole dB by an attacker. As we all know error are very important at the development phase of any web based application but we should always keep in mind that they should be disabled on Live Site or must be logged to file with limited access.

Union Based SQL Injection - It makes use of UNION SQL operator for combining the results of 2/more SQL statements into a single result which is returned as a part of HTTP response.

2. **Inferential SQL Injection** - This is also known as "**Blind SQLi**". It takes longer time to attack an dB because just like others, this category is too dangerous for vanishing the dB. This type of attack is referred as Blind SQLinjection attack because here data is not actuallytransferred via Web Application & thus attacker is not able to see the actual result. Here instead destroyer reconstructs the dB structure by observing Web Application responses & the behavior of dB server to that responses.The **2 types of attacks included in this category** are

Content Based - Also called as "**Boolean based Blind SQLi**". In this type of attack, SQL query is sent to dB which forces the running application to return the result based on the query whether it is **TRUE** or **FALSE**. The content of HTTP response can remain same or can vary depending on the query result. If the outcome of query turns out to be TRUE, the hacker slows down the dB (especially huge databases) since he needs to scan the dB character to character.

Time based SQL Injection - It is the technique where an SQL query sent to dB forces dB to wait for some amount of time span (usually in seconds) before it responds to particular query. The time taken helps attacker to indicate the result either in TRUE state or in FALSE state.If the outcome of query turns out to be TRUE, the hacker slows down the dB (especially huge databases) since he needs to scan the dB character to character.

3. **Out-of-Band SQL Injection** - This attack is the most uncommon attack that takes place as it depends on some of the features of dB server which need to be enabled that are used by Web Application. It usually takes place when destroyer is not able to access the same network channel as that of actual user. The alternative for this attack is the ties based SQLi especially when response of the server is not stable.

III – WORKING OF SQL INJECTION

In order to insert vulnerabilities against dB server, attacker must be able to get the inputs within the Web Application which will be in the form of SQL query. For perfect attack to take place, vulnerable website should include inputs of the user within SQL statement. Attacker will now be able to insert the payload which will be a part of SQL query & this now runs in the opposition of the dB Server.

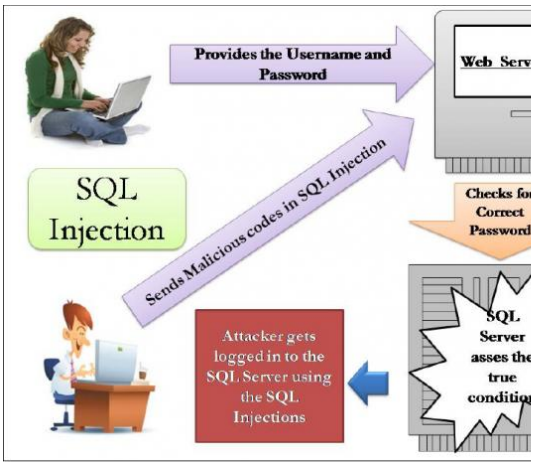


Fig. Overview of SQL Injection attack

IV – SQLi ATTACKS

SQLi **Mainly occurs** when -

1. Data is entered by the user from untrusted source.

2. Data is used dynamically to construct the SQL Query.

Main Consequences of SQLi are as follows -

Authentication - Sometimes poor SQL statements are use to cross check the username & password. So it may happen system may connect to another user without having the earlier knowledge of passwords.

Authorization - If authorization information is held in a SQL database, it may be possible to change this information through the successful exploitation of a SQL Injection vulnerability.

Integrity - As we can read sensitive data very easily, it is also possible to edit data (delete/modify) that content very easily with SQLi attack.

Confidentiality - Since SQL databases generally hold sensitive data, loss of confidentiality is a frequent problem with SQL Injection vulnerabilities.

SQLi attack involves insertion of malformed SQL query into a web application with the help of client-side input. The attack perverts the intentions of net programmers who write queries & provides input methods that can be exploited.

SQLi attacks includes "injection" / insertion of SQL query via input data from the user to the web Application. Successful SQLi can read sensitive information from dB, alter dB (Updating, Insertion, Deletion), can execute admin operations on dB (DBMS (Database Management System) shutdown), recovery of the file present in DBMS file. In few cases, it can also command OS (Operating System). SQLi attacks are the type of attacks where SQL statements are entered in the plane data input to affect the pre-defined SQL command.

V – SQL INJECTION DETECTION TOOL

SQLi destroys Web Application & code of database. To prevents this means we should resolve vulnerabilities present in the code. Dynamic code that generates SQL query using data from external source must be checked keenly. For very huge projects, using automatic source code scanning tools for scanning information.If you use any third-party applications that utilize a database back-end, it's vital that that you follow any vendor updates regarding vulnerabilities and patches to ensure the new code isn't introducing vulnerabilities into your own system.

Even though DBA (Database Admin) & developers of application follow the best practice to save data, they should deploy application layer firewall or WAF (Web Application Firewall).

Web Application Firewalls provide protection beyond TNF (Traditional Network Firewall) & prevention or detection system.Barracuda Network Incorporated (BNI) helps to prevent SQLi attacks, cross site scripting & other types of attacks that targets flaws in Application Logic / Technical Vulnerability in the software.

Best WAFs can detect techniques exploited by attacker using SQLi injection statement like obfuscating the attack by encoding the portion of injected commands.

The choosen Application Layer Firewall must allow to modify the traffic to the specific signal.SQL injection takes place via the URL-query string. We should regularly review our Web server's logs to look for anomalous queries that may be injection attempts.

SQL Injection Attacks are used either to gather or damage information held in dB & can also be used as launch pads to reach the depth of firm's system / network (private / public). By installing basic defences, user makes sure that his dB is quite harder to get leaked.

VI – TIPS TO PREVENT SQLi

The very simple way to check whether your site is susceptible to SQLi is to enter (')-Quote Character in the web form or in the query based URL to check what type of error message gets displayed. This quote character will confuse the dB, if not handled properly.

SQLi needs tackling in many levels. But most importantly it is to be used with stored procedures of parameterized type because here user requests are made with the help of parameters & user defines sub routines rather than constructing SQL statements directly by the user. This passed parameters are not only Type Safe (Strict Format) but they also decrease the possibility of SQL injection attack. If in case, access to the dB is permitted via Stored Procedures, then there is no need to explicitly grant user the permission to any tables within dB.

As said parameterized stored procedures reduces the risk of SQLi attacks, application of user still needs validation & must sanitize input data (i.e. whether entered data is supplied by the authorized& authenticated user, or is read by appropriate cookie). It means whether the information entered by the user is of correct type, proper format, defined length & most importantly is it within an expected range. The data which is not formatted well & is incorrect must be rejected immediately.

The thing to remember always is web application should not be run with admin priorities at Server / Database Level or else destroyer potentially could the task of modifying data which only Admin has rights. Web Application should always run on server with minimum privileges, only those privileges which are necessary for it to function on network & dB permission should be set only to the resources that is really essential. With this attacker is confined with only limited set of permissions & so attack can be minor without causing large amount of loss to the dB.

Large number of web applications sanitize their inputs by using known unsafe special character like "<" (less than symbol) or "/" (escape symbol). But however, this should not be done, because users who are willing to attack the data can usually find the alternative means to get the result. Attackers can easily do this by the mechanism known as "Character Encoding". Instead, code must go for checking the safe input. This type of Validation must be implemented on trusted server end & not on client end. With such kind of validation at very first attempt of validation only data must enter the dB or particular script. These checks also apply to data received from internal applications or entered or edited by internal users (for this, assumption of data coming from untrusted end should be made).

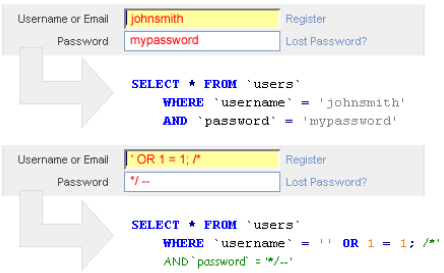
VII – AUTOMATED SQLi DETECTION

Before exploiting SQLi vulnerability in an automated manner, it should be detected first. But detecting this vulnerability in an automated fashion is not the good choice because these automated tools take heavy amount of time as they enumerate amount of payloads which are by default. At present, SQL Map gives you opportunity to limit the number by selecting only required payloads like -risk, -level, -technique, -dbms, etc... Even if this is utilized, we still don't have completely grained control over the payloads which were sent & as well were used for detection method implementation. Secondly, these automation tools are not fool proof always. Lastly, when web application utilizes something out of the ordinary such as a custom CSRF mitigation implementation / an exotic authentication method, this tools usually won't be able to reach all points of potential injection.

VIII – STEPS OF SQL INJECTION

1. User presents the form to the server which is adopted by the attacker in between.
2. Attacker absorbs the file & gets the attack on the form data.
3. Application is then forward to the dB in the form of simple SQL query statement.

4. Database runs query containing attackers input values & sends encrypted results back to the user.
5. These encrypted file is absorbed by the attacker from the server.
6. Attacker decrypts data as normal & sends data to the actual user.



IX – CONCLUSION

This paper describes the issues of thedata that gets corrupted when taking the transfer from one end to another end via internet. Also how to handle the data safely without getting hacked & how to use the strong credentials to save our data from getting lost. Various prevention mechanisms help to develop the awareness to save the dB from outsiders who destroys our dB & also sometimes the entire OS if accessible.Using these metrics, a true Adaptive Database Firewall, based on behavioral analysis, is vastly superior to a WAF in identifying SQL injection attacks.This is true because an Adaptive Database Firewall can be trained quicker, has minimal false positives, and is capable of seeing through attack obfuscation techniques which slip easily through WAFs.

X – ACKNOWLEDGEMENT

We thank our colleagues from IMCOST who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper.

We thank our faculty Mrs. Reeta Singh for assistance provided to us by proper IEEE formats and all faculties for guiding us.

XI – REFERENCES

1. https://www.arneswinnen.net/2013/09/automated-sql-injection-detection
2. https://en.wikipedia.org/wiki/SQL_injection
3. https://cdn.tutsplus.com/net/uploads/legacy/Articles/009_Security/NETTUTS-SEC/mainpages/flow%20a%20-%20sql%20inject.gif
4. http://www.w3schools.com/sql/sql_injection.asp
5. http://www.computerweekly.com/tip/SQL-injection-detection-tools-and-prevention-strategies
6. SQL Injection Attacks & Defense - Justin Clarke