Database security: SQL Injections

Josh Russell

East Carolina University

Abstract

In this paper I will be discussing the topic of securing databases. The importance of securing servers with databases, is that there is a lot of person or sensitive data store on it. In today's world, almost everything is stored online. People pay bills and shop on different sites. Sites like amazon that store your credit card number and home address stored on the site. All this information is stored on a database. Securing these devices is very important to many organizations and companies. I will be talking about the top ways to secure a database server. We will be discussing monitoring threats that face web application with database. Some of the top threats like Malware, SQL injection attacks, and user error. This paper will cover the basic ways to secure and defended against these attacks.

Database security: SQL Injections

In today's day and age, whether its logging onto social media or going to the doctors, your information is stored on a database. Database is the way the in which to store and organize a whole bunch of data. The database is stored on a server using a system called a Relational Database Management System or RDBMS. The basic language for RDBMS is SQL which stands for Structured Query Language. The most popular types of RDBMS mostly used today is MS SQL Server, IBM DB2, and MySQL.

What is SQL

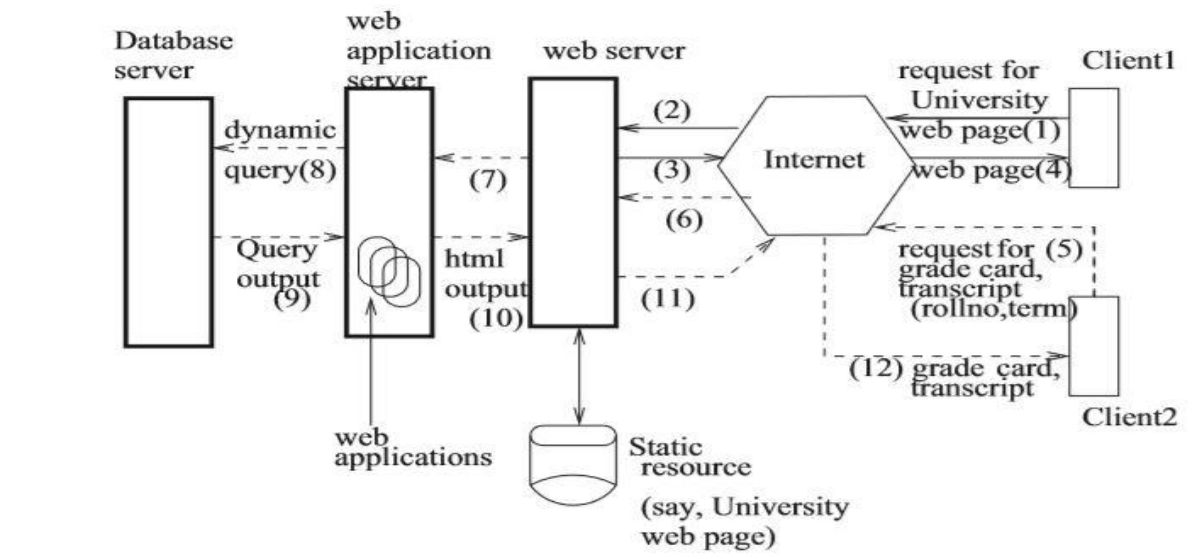
SQL is a programming language that can create, delete, modify data within a database. This is very helpful in all industries. In a busy world, employees need to have a way to access data and organize that data into a readable form. A database is what holds tables. Tables contains a bunch of related data in rows and columns. The table is held up with data field, the fields are headers for the columns. The rows in the tables are known as records. There are many different types of commands to retrieve certain data. The SELECT command is where you pull information from a database. For example, a command "SELECT Baseball FROM sports;" the admin is selecting the all the data in the field Baseball from the data table sports. When setting up a web application there are two sides involved. The first side is the front end. The front end is the web interface, that is the part that the user will interact with. The front-end interface contains the log in and any contented that the website contains. This is written with HTML and CSS code. The HTML is the back bone of the website and the CSS is what gives the web application its style. The back end of the web application is where the database is stored. The server with this application will have a type of PHP scripts. PHP is a server-side language that communicates the front end to the back end. This is where the admin can interact with the database. Using SQL in a database system is a powerful tool. A system admin or hacker can manipulate data in a database in different ways to perform SQL injections or SQLi.

History of SQLi

One of the most common ways to hack a database is by using one of the easiest types of hacks. (Cox) This type of hack is where the attacker will put a set of key SQL commands to get key information about the database. One of the first documented SQLi attack was in 1998 by Jeff Forristal or also known as rain.forest.puppy as his alias. Forristal is now the CTO of Bluebox security. While in college in December of 1998, Forristal found a vulnerability with SQL database that was not yet well known. In the 1990 not, many web sites were using full functional SQL database, but were instead using just a Microsoft Access database application. Forristal said “I can completely change the way SQL works” and “At that point, there were no real security properties fronting a database.” (Kerner) In the late 1990, there was no threat to database at that time. Which is kind of strange knowing that this is where a lot of personal information is keep. I would want that to be secure. SQL injections are still a major concern in todays IT world. SQL injections are constantly landing as one for the biggest security vulnerabilities. In an article from Motherboard says, “ Today, over 15 years after it was first publicly disclosed, SQLi repeatedly sits at the number one spot of vulnerabilities in the OWASP Top 10 report, which is released every three years by the Open Web Application Security Project (OWASP).” (Cox)

Set up of Web Servers

When setting up a web server that has a database for that web server the system admin needs to make sure that SQL injections are not possible. This can be a difficult task to do because the hack is growing and becoming more sophisticated. One way to help defend against Injections is to set up the web server with multiple tiers application. In an article titled “Defeating SQL injection attack in authentication security: an experimental study", they gave a scenario with a schools web server. The web server was set up with three tiers; web server, web application server, and database server. (Roy)



This was to add another lay of security so that a SQL injection would not produce an error with information about the database. There are two types of detection techniques used with in a SQL database. One is static or pre-generated detection, this is a method that handles the identification of vulnerabilities in application programs that can be misused by an attacker. This all depends on how well of the input validity checking module is set up. This is the module that checks the input form the users to make sure it is a legit login or query request. The second one it dynamic or post-generated. This is based on how effective the run-time module is set up. What is module does is match a set of query strings to set if it is a SQL injection attack.

Defending against SQL

As SQL injections are one of the oldest but easier attacks for someone to do, database administrators need to find the best way to defend against this attack. There are three types of defense strategies to take when defending a database. They are defense coding, SQLIV detection, and SQLIA runtime prevention.

The first type of defense is defense coding. This is a straightforward solution to SQL injections. Some defense practices are parameterized queries or stored procedures, this is where the admin would replace dynamic queries with properly coded parameterized queries or it stores the procedures. This forces developers to set the SQL codes structures. The next step is to escape. This is were if the SQL query is invalid the escape procedures will use the standard escape libraries instead of custom escaping method. This will help not give any information about the database away. The last step in defense coding is doing the data type validation. This works with the escape method to see if it is a mismatch query and to see if it needs to reject the query.

In the SQLIV detection, there have been many researchers have come up and developed different methods to find SQLIV. The first one is Code-based vulnerability testing. This is aimed to generate a test to find vulnerabilities in the program points. There is also a MUSIC tool or Mutation-based SQL injection vulnerability checking. This was developed by Jose Fonseca, Marco Vieira, and Henrique Madeira for automatically injecting SQLIVs into web programs and generate SQLIAs.

Another method of SQLIV is concrete attack generation. This is the last and greatest method of generating test inputs that expose SQLIVs in a web program. One powerful tool used in SQLIV is called SUSHI. These solves path conditions that lead to SQL statements and takes out test inputs containing SQLIAs from the solution pool. A SQLIA runtime is a way to check current runtime against legitimate quires. One type of SQLIA is called Learning-based prevention. These is set up between the application server and the database server. This program checks all the queries for key SQL words to see if they are legitimate or malicious.

SQL Injection Attacks

To look at a live attack in 2011, Sony Pictures face a SQL injection. There was a hacking group called Lulz Security. The two men charged was a homeless man in San Francisco and Arizona native Cody Kretsinger. (Martin) The attackers release personal information like username, passwords, emails, and addresses. After this was reported the security firm Imperva, found that they two had to face against the SQL injections. They saw that there web applications faced 71 attacks an hour. It appeared that Kretsinger used a proxy site the try and mask his address. The site was called hidymyass.com but thought to not do a good job. Kretsinger face 15 years in prison for the Sony breach.

Conclusion

In conclusion SQL injections are a very basic and simple attack. The SQL injection is still one of the top attacks that face network enterprise databases. When being a database administrator, one needs to make sure that they do they proper steps. There are many different methods to setting security for the database. One is to sanitize the code by using defense coding. There is also setting up SQLIV detections or SQLIA runtimes. SQL servers are a big part in a large organization and we need to know the best ways to protect them.

References

Cox, J. (2015, November 20). The History of SQL Injection, the Hack That Will Never Go Away. Retrieved from https://motherboard.vice.com/en_us/article/aekzez/the-history-of-sql-injection-the-hack-that-will-never-go-away

How Was SQL Injection Discovered? (n.d.). Retrieved from https://www.esecurityplanet.com/network-security/how-was-sql-injection-discovered.html

Martin, A. (2013, October 30). LulzSec's Sony Hack Really Was as Simple as It Claimed. Retrieved from https://www.theatlantic.com/technology/archive/2011/09/lulzsecs-sony-hack-really-was-simple-it-claimed/335527/

Roy, S. (2011). Detecting and Defeating SQL Injection Attacks. *International Journal of Information and Electronics Engineering*. doi:10.7763/ijiee.2011.v1.6

Das, D., Sharma, U., & Bhattacharyya, D. K. (2017). Defeating SQL injection attack in authentication security: An experimental study. *International Journal of Information Security, 18*(1), 1-22. doi:10.1007/s10207-017-0393-x