MS-ISAC

MS-ISAC Security Primer

SQLi

March 2017, SP2017-1019

Web applications are database driven in order to store user preferences, personally identifiable information, and other sensitive user information so that the web pages can build customized content for each user. The web application communicates with the database using Structured Query Language (SQL), a programming language for managing databases that allows a user to read and manipulate the data. Malicious cyber actors exploit the relationship between a web application and a database via SQL injection (SQLi) attacks, in which a SQL command is typed into a web form entry field in an attempt to access the information stored in the database. If the SQLi vulnerability is successfully exploited, a malicious cyber actor could view, exfiltrate, modify, delete, or corrupt the information stored in the database.

RECOMMENDATIONS:

- Conduct a vulnerability scan of Internet-facing applications, focusing on identifying and remediating SQLi vulnerabilities and patching out-of-date software, especially content management systems.
- Consider implementing a Web Application Firewall on the affected web server for an additional layer of protection.
- If your website is hosted by a third party, establish a relationship with your hosting provider and have its contact information readily accessible in case of a compromise. Ensure that the hosting provider conducts regular vulnerability scans and updates the website to address vulnerabilities.
- Create custom, general error messages for the web application to generate, as malicious cyber actors can gain valuable information, such as table and column names and data types, through default error messages generated by the database during a SQLi attack.
- Validate user input prior to forwarding it to the database. Only accept expected user input and limit input length. This can be done by implementing a whitelist for input validation, which involves defining exactly what input is authorized.
- Implement the principle of least privilege for database accounts. Administrator rights should never be assigned to application accounts and any given user should have access to only the bare minimum set of resources required to perform business tasks. Access should only be given to the specific tables an account requires to function properly.
- The database management system itself should have minimal privileges on the operating system, and since many of these systems run with root or system level access by default, it should be changed to more limited permissions.
- Isolate the web application from the SQL instructions. Place all SQL instructions required by the application in stored procedures on the database server. The use of user-created stored procedures and prepared statements (or parameterized queries) makes it nearly impossible for a user's input to modify SQL statements because they are compiled prior to adding the input. Also, have the application sanitize all user input to ensure the stored procedures are not susceptible to SQLi attacks.
- Use static queries. If dynamic queries are required, use prepared statements.
- For additional information, refer to the MS-ISAC White Paper on SQLi, available on the MS-ISAC website.

The MS-ISAC is the focal point for cyber threat prevention, protection, response, and recovery for the nation's state, local, tribal, and territorial (SLTT) governments. More information about this topic, as well as 24x7 cybersecurity assistance for SLTT governments, is available at 866-787-4722, SOC@cisecurity.org, or https://msisac.cisecurity.org/. The MS-ISAC is interested in your comments - an anonymous feedback survey is available at: https://www.surveymonkey.com/r/MSISACProductEvaluation.