Impact Factor (SJIF): 4.542

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 4, Issue 3, March-2017

SQL Injection Analysis, Detection, Exploitation and Report Generation

Zufeshan J. Ansari, Karan R. Kapoor, Rohit A. Singh, Prof. Prashant H. Rathod

Abstract*— *In today's modern world everything have become digital. Now everything is dependent on websites via the internet. Every user is dependent on a relevant application, and every application has a database, what if the database is vulnerable to exploitation, typically a sequel injection. "SQL injection is a code injection technique, used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution." Let's say that you are auditing a web application and found a web page that accepts dynamic user-provided values via GET, POST or Cookie parameters or via the HTTP User-Agent request header. You now want to test if these are affected by SQL injection vulnerability, and if so exploit then to retrieve as much information as possible from the back-end database management system, even be able to access the underlying file system and operating system.

Developing an SQL Injection tool to detect, exploit and form a report to state the level of vulnerability in the web application. Using Python language as a platform and Tkinter as a framework to develop GUI. Whole procedure (action) will be automated by the help of this tool called as "SQL-I AUTO". Admin Finder, Dynamic IP changing (being Anonymous) over the network, information gathering of the site owner will be some additional features of this tool. SQL-I AUTO will 1) Identify the vulnerable parameters. 2) Identify which SQL injection techniques can be used to exploit the vulnerable parameters. 3) Fingerprint the back-end database management system. 4) Providing IP hiding facilities over the network.

Index Terms- **Anonymous, Cybercrime, Database, Security professional tool, SQL injection, Vulnerability.**

I. INTRODUCTION

Computer security, also known as a cyber security or IT security, is the protection of information system from theft or damage to the hardware, the software, and to the information on them, as well as from disruption or misdirection of the services they provide. The vulnerability is a system susceptibility or flaw. It includes controlling physical access to the hardware, as well as protecting against the harm that may come via network access, data and code injection, and due to malpractice by operators, whether intentional, accidental or due to them being tricked into deviating from secure procedures. The field is of growing importance due to the increasing reliance on computer systems and the Internet in most societies, wireless networks such as Bluetooth and Wi-Fi and the growth of "smart" devices, including smartphones, televisions and tiny devices as part of the Internet of Things. To secure a computer system, it is essential to understand the attacks that can be made against it.

II. WHAT IS SQL INJECTION ATTACK?

SQL Injection [13] is a type of web application security vulnerability [5] in which an attacker can submit a database SQL command, which is executed by a web application, exposing the back-end database. SQL Injection attacks [7][10] can occur when a web application utilises user-supplied data without proper validation or encoding as part of a command or query. The specially created user data tricks the application into executing unintended commands or changing data. SQL Injection lets an attacker to create, read, update, alter, or delete data stored in the back-end database. In its most common form, SQL Injection permits attackers to access sensitive information such as social security numbers, credit card number or other economic data. According to Vera code's State of Software Security Report, SQL Injection is one of the most widespread types of web application security vulnerability.

Key Concepts of SQL Injection

- SQL injection is a software vulnerability [6] [14] that occurs when data entered by users is directed to the SQL interpreter as

a part of an SQL query.

- Attackers provide specially crafted input data to the SQL interpreter and trick the interpreter to execute unintended commands.
- Attackers utilise this vulnerability by providing specially crafted input data to the SQL interpreter in such a manner that the interpreter is not able to distinguish between the intended commands and the attacker's specially crafted data. The interpreter is misled into executing unintended commands.
- SQL injection exploits security weaknesses at the database layer. By exploiting the SQL injection flaw, attackers can create, modify, read, or delete sensitive data.

III. TYPES OF SQL INJECTION

SQL Injection can be categorized [1] into three major categories – In-band SQLi, Inferential SQLi and Out-of-band SQLi.

A. In-band SQLi (Classic SQLi): In-band SQL Injection is the most common and simple to exploit of SQL Injection attacks. In-band SQL Injection occurs when an attacker can use the same communication channel to both launch the attack and gather results.

The two most common forms of in-band SQL Injection are Error-based SQLi and Union-based SQLi.

1) Error-based SQLi: Error-based SQLi is an in-band SQL Injection technique that depends on error messages thrown by the database server to obtain information about the structure of the database. In some cases, error-based SQL injection alone is enough for an attacker to enumerate an complete database. While errors are very useful during the development phase of a web application, they should be deactivated on a live site or logged to a file with restricted access instead.

2) **Union-based SQLi:** It influences the UNION SQL operator to combine results of two or more SELECT statements into a single result which is then returned as part of the HTTP response.

B. Inferential SQLi (Blind SQLi): Inferential SQL Injection, unlike in-band SQLi, may take longer for an attacker to exploit; however, it is just as dangerous as any other form of SQL Injection. In an inferential SQLi attack, no data is transferred via the web application, and the attacker would not be able to see the result of an attack in-band (which is why such attacks are commonly referred to as "blind SQL Injection attacks"). Instead, a Blind-time-based SQLi attacker can reconstruct the database structure by sending payloads, observing the web application's response and the resulting behaviour of the database server.The two types of inferential SQL Injection are Blind-Boolean-based SQLi and Blind-Time-based SQLi.

1) Boolean-based (content-based) Blind SQLi: Boolean-based SQL Injection is an inferential SQL Injection technique that depends on sending an SQL query to the database which forces the application to return a different result depending on whether the query returns a TRUE or FALSE result.

Depending on the result, the data within the HTTP response will change, or remain the same. This permits an attacker to infer if the payload used returned true or false, even though no data from the database is returned. This attack is usually slow (especially on large databases) since an attacker would need to enumerate a database, character by character.

2) Time-based Blind SQLi: Time-based SQL Injection is an inferential SQL Injection technique that relies on sending an SQL query to the database which forces the database to wait for a specified amount of time in seconds, before responding. The response time will specify the attacker whether the result of the query is TRUE or FALSE.

C. Out-of-band SQLi: Out-of-band SQL Injection is not very common, mostly because it relies on features being enabled on the database server being used by the web application. Out-of-band SQL Injection occurs when an attacker can't use the same channel to launch the attack and gather results. Out-of-band techniques offer an attacker an alternative to inferential time-based techniques, especially if the server responses are not very stable (making an inferential time-based attack unreliable).

Out-of-band SQLi techniques would depend on the database server's ability to make DNS or HTTP requests to deliver data to an attacker. Such is the case with Microsoft SQL Server's xp_dirtree command, which can be used to make DNS requests to a server attacker controls; as well as Oracle Database's UTL_HTTP package, which can be used to send HTTP requests from SQL and PL/SQL to a server an attacker controls. SQL Server will now proceed to list all the folders from \\test.attacker.com\. To do this, it must first resolve the address of the domain test.attacker.com, for which it makes a DNS query to attacker's DNS server. The attacker can monitor DNS server logs and look for queries to test.attacker.com. If such a DNS query is made, it means that the SQL Injection vulnerability is exploitable via an Out-of-Band vector.

IV. PROPOSED APPROACH

The user will enter an URL that will detect through an algorithm whether it is vulnerable or not, If URL is vulnerable to SQL Injection then the exploitation will carry out till maximum data is extracted, Else if URL is not vulnerable, then the application will prompt the user to re-enter another URL. ([3],[12],[15],[16])

As compared to other SQL-I tools our tool is also an SQL-I exploit tool, but having some additional features then other such as Information Gathering, Admin Finder, Anonymous (being hidden over the network whenever an attack is on), having huge database support. The name suggests that our tool is fully Automated not partially automated as compare to other tools. It supports both HTTP and HTTPS websites. You can perform SQL injection via GET, POST or cookies. It also supports authentication (Basic, Digest, NTLM HTTP authentications) to perform a SQL injection attack. The tool supports a wide range of database servers including MySQL, Oracle, PostgreSQL, Microsoft SQL Server, and Microsoft Access.

A. Admin Finder: In Admin Finder section there will be stored text files containing Google dorks, it will redirects the current user to admin interface by automatically including username as well as password. It will target a typical vulnerability i.e. Broken Authentication.

B. Upload Script File: This module allows the user to upload their script, which will be helpful to carry on more power attack on the web application.

C. Report Generation Module: This module allows the user to generate report on the level of Vulnerability. Which describe the level of vulnerability in the form of low, medium, high set flags

V. COMPARATIVE ANALYSIS [17]

BSQL Hacker: BSQL Hacker is a nice SQL injection tool that helps you perform a SQL injection attack against web applications. This tool is for those who want an automatic SQL injection tool. It is especially made for Blind SQL injection.

Key features

- Can automate most of the new SQL Injection methods those relies on Blind SQL Injection.
- Console and GUI Support.

Advantages

- BSQL Hacker SQL injection tool supports MSSQL and ORACLE only.
- BSQL Hacker aims for experienced users as well as beginners who want to automate SQL Injections (especially Blind SQL Injections)

Disadvantage

- It is incompatible with huge databases.

SQL Map: SQL Map is the open source SQL injection tool and most popular among all SQL injection tools available. This tool makes it easy to exploit the SQL injection vulnerability of a web application and take over the database server. It comes with a powerful detection engine which can easily detect most of the SQL injection related vulnerabilities.

Key features

- Full support for MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB, HSQLDB and Informix database management systems.
- Support to directly connect to the database without passing via a SQL injection, by providing DBMS credentials, IP address, port and database name.

Advantages

- Support huge database.
- Especially for Error-based and Blind-based

Disadvantage

- Does not provide Anonymous attack.

SQL Ninja: SQL ninja is a SQL injection tool that exploits web applications that use a SQL Server as a database server. This tool may not find the injection place at first. But if it is discovered, it can easily automate the exploitation process and extract the information from the database server. This tool can add remote shots in the registry of the database server OS to disable data execution prevention. The overall aim of the tool is to allow the attacker to gain remote access to a SQL database server.

Key features

- Creation of custom xp_cmdshell if the original one has been removed.
- Data extraction, time-based or via a DNS tunnel.

Advantages

- It can run on any UNIX based platform with a Perl interpreter.
- It supports a wide range of operating systems such as Linux, FreeBSD, Mac OS X, iOS.

Disadvantage

- SQLninja does not support Windows operating system.

Safe3 SQL Injection: Safe3 SQL Injector is another powerful but easy to use SQL injection tool. Like other SQL injection tools, it also makes the SQL injection process automatic and helps attackers in gaining the access to a remote SQL server by exploiting the SQL injection vulnerability. It has a powerful AI system which easily recognises the database server, injection type and best way to exploit the vulnerability. It supports both HTTP and HTTPS websites. You can perform SQL injection via GET, POST or cookies. It also supports authentication (Basic, Digest, NTLM HTTP authentications) to perform a SQL injection attack.

Key Features:

- Support to enumerate databases, tables, columns, and data.
- Support to ip domain query, web path guess, md5 crack, etc.

2.5 SQL Sus

SQL Sus is another open source SQL injection tool and is a MySQL injection and takeover tool. This tool is written in Perl, and you can extend the functions by adding your codes. This tool offers a command interface which lets you inject your SQL queries and perform SQL injection attacks. This tool claims to be fast and efficient. It claims to use a powerful blind injection attack algorithm to maximise the data gathered.

Key features

- Support for GET and POST parameters injection vectors
- Support for HTTP proxy and HTTP simple authentication.

Advantages

- It supports both quoted and numeric injection
- It discovers the exact injection space going through all possible restriction (web server, Suhosin patch...), to inject as much as possible at once.

Disadvantages

- It does not dump data into XML format.

2.6 Mole

Mole or (The Mole) is an automatic SQL injection tool available for free. You only need to find the vulnerable URL and then pass it in the tool. This tool can detect the vulnerability from the given URL by using Union based or Boolean based query techniques.

This tool offers a command line interface, but the interface is easy to use. It also offers auto-completion on both commands and command arguments. So, you can easily use this tool. This tool was written in Python and requires only Python3 and Python3-lxml.

Features

- Support for MySQL, Postgres, SQL Server and Oracle.
- Automatic SQL injection exploitation using union technique.

Advantages

- Automatic blind SQL injection exploitation.
- Exploits SQL Injections in GET/POST/Cookie parameters.

Disadvantages

- It supports a limited range of database management systems
- It does not execute arbitrary commands.

2.7 SQL Brute

This tool is useful in brute extraction of data from a database through vulnerable SQL injection weaknesses. This tool is developed in Python, utilises multithreading and need standard libraries. SQLBrute is a tool for brute forcing data out of databases using blind SQL injection vulnerabilities. It supports time-based, and error based exploit types on Microsoft SQL Server, and error based exploit on Oracle.

Key features

- It supports time based, and error based exploit types on Microsoft SQL Server, and error based exploit on Oracle.
- It uses multi-threading and doesn't require non-standard libraries.

2.8 BobCat

BobCat is a tool to aid an auditor in taking full advantage of SQL injection vulnerabilities. It is based on AppSecInc research. It can list the linked severs, database schema, and allow the retrieval of data from any table that the current application user has access to. This tool is useful in helping auditors exploit SQL injection weaknesses. BobCat relies on rears by AppSecInc. This scanner can display database schema, linked servers and enable data retrieval from a table that a user of the present application can access.

Key features

- It can exploit SQL injection bugs/opportunities in web applications, independent of language.

2.9 Havij SQL Injection

Havij is an automated SQL Injection tool that helps penetration testers to find and exploit SQL Injection vulnerabilities on a web page. The power of Havij that makes it different from similar tools is its injection methods. The success rate is more than 95% at injecting vulnerable targets using Havij. The user-friendly GUI (Graphical User Interface) of Havij and automated settings and detections makes it easy to use for everyone even amateur users.

Key feature

- Dumping the data to a file and can be stored in XML format.

It supports both HTTP and HTTPS

After analysing the tools that are currently running in the market, we came across certain drawbacks that we overcome in our tool SQLi Auto.

Table 1: Comparison among various tools

TOOLS	DATABASE	TYPE SPECIAL	AUTOMATED	ANONYMOUS	BYPASS FUNCTION	INFORMATION GATHERING
BSQL Hacker	MsSQL, ORACLE.	Blind	Partially	No	No	No
SQL Map	ORACLE, MySQL, ..etc	Error + Blind	Partially	No	No	No
SQL Ninja	MsSQL.	Error	Partially	No	No	No
SQLSus	MySQL.	Union	Partially	No	No	No
Sfe3 SQL	MySQL, MsSQL	Blind	Partially	No	No	No
Mole	MYSQL,MsSQL, Postgres.	Union + Boolean	Partially	No	No	No
SQLi-Auto	MsSQL,MySQL ,Oracle, Postgres, DB2, IBM.	Error + Union + Boolean +Blind	Fully Automated	Yes	Yes	Yes

VI. CONCLUSION

As today there is a need for security, user privacy plays an important role in any organisation, the keen role of an organisation is to protect data of the user. So this tool might be helpful to detect [8] the vulnerability in the application as fast as possible.

The tool might be very useful for a security researcher to identify the typical sequel injection in an application. The main purpose of this tool is to detect, exploit and find the level of vulnerability in the given web application. Beneficial for Security researcher that the whole task will do automatically. Even the tables name will be automatically added to find the admin password. The tool will be user-friendly so that anyone can use it to exploit the web application which is vulnerable to sequel injection attack.

The key to success is to detect vulnerability and exploit it.

REFERENCES

[1] Halfond, W., Viegas, J., & Orso, A. (2006). "Classification of SQL Injection Attacks and Countermeasures." SSSE 2006

[2] Sushila Madan and Supriya Madan, ―Security Standards Perspective to Fortify Web Database Applications From Code Injection Attacks,‖ IEEE International Conference on Intelligent Systems, Modelling and Simulation, page 226-230, 2010.

[3] E. Bertino, A. Kamra, and James P. Early, ―Profiling Database Applications to Detect SQL Injection Attacks,‖ IEEE Conference, 2007.

[4] M. Shema, Seven Deadliest Web Application Attacks, Elsevier Inc., 2010, pp. 47-69.

[5] The Open Web Application Security Project (OWASP), https://www.owasp.org/index.php/Top_10_2010-Main.

[6] Xie, Y., and Aiken, A. Static detection of security vulnerabilities in scripting languages. In USENIX Security Symposium (2006).

[7] Sandeep Nair Narayanan, Alwyn Roshan Pais, & Radhesh Mohandas. Detection and Prevention of SQL Injection Attacks using Semantic Equivalence. Springer 2011

[8] H. Shahriar and M. Zulkernine, ―MUSIC: Mutation-based SQL Injection Vulnerability Checking,‖ The Eighth International Conference on Quality Software, IEEE Computer Society, 2008.

[9] M. Shema, Seven Deadliest Web Application Attacks, Elsevier Inc., 2010, pp. 47-69.

[10] Halfond, W. G. J. and A. Orso (2005). AMNESIA: Analysis and monitoring for Neutralizing SQL-injection attacks. ASE'05. Long Beach, California, USA.

[11] Shaukat Ali, Azhar Rauf, and Huma Javed, 2009. "SQLIPA: An Authentication Mechanism Against SQL Injection," European Journal of Scientific Research, ISSN 1450-216X Vol.38 No.4, pp 604-611.

[12] G.Anil Kumar, Srinivas Baggam, Enhanced Model of SQL Injection Detecting and Prevention, International Journal of Science and Advanced Technology (ISSN 2221-8386) Volume 1 No 9 November 2011

[13] MayankNamdev, FehreenHasan, Gaurav Shrivastav "Review of SQL Injection Attack and Proposed Method for Detection and Prevention of SQLIA"Volume 2, Issue 7, July 2012

[14] Manas Kumar1, S. Senthil kumar2 and D. Sarvanan SQL INJECTION MONITORING SECURITY VULNERABILITIES IN WEB APPLICATIONS International Journal of Information Technology Volume 2, Issue 3, March 2014.

[15] Xiang Fu, Xin Lu, Boris Pelts verger, Shijun Chen "A Static Analysis Framework of Detecting SQL Injection Vulnerabilities" IEEE Transaction of computer software and application conference 2007

[16] Kontantinos kemalis and Theodoros Tzouramanis "Specification-Based approach on SQL Injection Detection" ACM 2008.

[17] SQL Injection Detection and Prevention Tools Assessment By Atefeh Tajpour CASE Center University Technology Malaysia Kuala ,Lumpur, Malaysia ; Mohammad Zaman Heydari ,IT & Management Dep UCSI University Kuala Lumpur, Malaysia ; Maslin Volume 2, Issue 6, June 2012 www.ijarcsse.com © 2012, IJARCSSE All Rights Reserved Page | 46 Masrom ,CASE Center University Technology Malaysia Kuala ,Lumpur, Malaysia , Suhaimi Ibrahim ,CASE Center UTM University Kuala Lumpur, Malaysia