DOI: 10.15439/2017R66
ACSIS, Vol. 10 ISSN 2300-5963

Analysis of SQL Injection Using DVWA Tool

Gajanan Shinde
Department of Information Technology,
PCCOE
Pune, India
gajushinde8046@gmail.com

Sandhya S. Waghere
Department of Information Technology,
PCCOE
Pune, India
sandhyawaghere@gmail.com

***Abstract*—As the World Wide Web has been constantly evolving, many industrial sectors, such as social networking online shopping, e-government and e-banking, they have made their services available on the web. However, this causes malicious attackers makes a main target on Web. SQL Injection is one of the most vulnerable attack. With the help of authenticated user input parameters to change the query's logic hacker insert some SQL character in SQL Statement. When request is produced from client end query is produced. Query have to handle before execution, because client input originates from external as well as it is malicious. Currently security researchers proposed different types of solutions to defeat SQL injection attack. One of the very dangerous web application is Damn Vulnerable Web application (DVWA). There is numerous data inside DVWA to learn beginner. DVWA likewise utilized as a kind of perspective to secure coding, application against SQL Injection is secured if developer is not exactly beyond any doubt about it.**

***Index Terms*—DVWA tool; SQL Injection; vulnerabilities;**

I. Introduction

Now a days Life is very easy with the help of web application. For activities, need to have some client contribution in web application. In a client function of web application, there are different malicious action. With the help of free access of web application, it is conceivable to attempt mischievous activity. By injection malicious code the attack is performed by abuse of input vulnerabilities. [9]. Right now, SQL Injection (SQLI) attack exploit most hazardous security vulnerabilities in different well known web applications i.e. Google eBay, Twitter, Facebook and so forth [10]. SQL Injection is one of the most vulnerable attack. With the help of authenticated user input parameters to change the query's logic hacker insert some SQL character in SQL Statement. When request is produced from client end query is produced. Query have to handle before execution, because client input originates from external as well as it is malicious. [2].

Steps for DVWA tool with connection of XAMPP.

1. Install XAMPP and DVWA
2. Copy DVWA Folder to XAMPP/htdocs
3. Open xampp control panel and start Apache, MySQL and Tomcat services
4. Open localhost/dvwa in any web browser
5. Login and do the procedure as flowchart

Figure 1 shows the proper flowchart of the result which gives the collection of malicious queries.

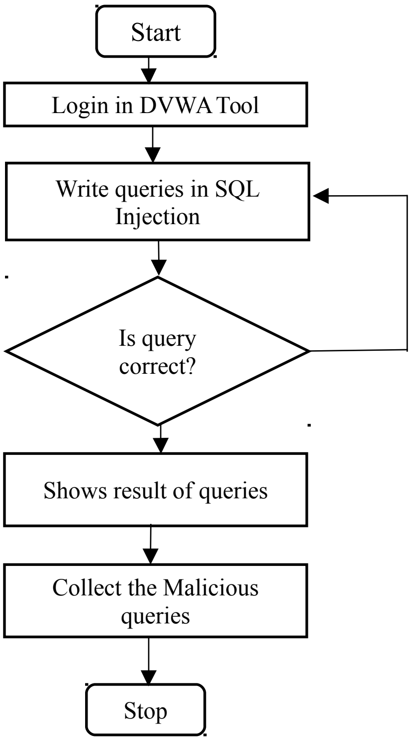


Fig1: Flow chart of malicious queries with the help of DVWA

A. Types of SQL Injection Attack

1) Tautologies based Attack
Attack Intent: Bypassing verification, finding insertable variables, separating information.
Description: when statement appears as true, malicious code at least contain one conditional statement, it is the main reason of tautology attack. Behavior of resultant query used in web application gives the outcome of these attack. The most general uses are to extract data with the help of bypass verification pages. In this sort of injection, WHERE condition is utilized by the hacker is to exploits an injectable field. Hacker need to know all coding development and vulnerable parameter to get the vulnerable result. Commonly at least one record to be returned so that the attack is effective.
Example: SELECT userDetails FROM person
WHERE loginId= ‘or 1=1 -- AND pass= AND pin= ”;

To activate tautology attack, we need to complete Conditional (or 1=1) logic in the SQL statement ,because it shows true result, then whatever query is written after that may have true result and attack is generated. [1].

2) Illegal Queries/Logically Incorrect Queries

Attack Intent: finding insertable variable, performing database finger-printing, separating information.
Description: This attack is used to get essential information from back end database. It shows error pages returned in application server.in simple fact hacker shows vulnerable parameters and error messages are created. Hacker tries to insert syntax conversion, type conversion and logical error into database.
Example: This attack is used to bring out the important data with the help of type conversion error.
Hacker insert some vulnerable parameter to existing code as follows:

"convert(int,(select top 1 name from sysobjects where xtype='u'))". The resulting query is:
SELECT accounts FROM users WHERE login='' AND pass='' AND pin= convert (int,(select top 1 name from sysobjects where xtype='u'));

In this attack hacker tries to extract first user table with the help of select query from the database.
The query tries to convert this table name into an integer. Since this is not a legitimate sort conversion, the database throws an error. In this attack there are two propose, with the help of error message hacker can see the database and the type conversion to occur caused by the error message which shows the value of string. [1].

3) Union Query

Attack Intent: Bypassing verification, separating information.
Description: In this attack we can use Union select query and try to convert normal query into vulnerable query. In this section hacker can do the vulnerabilities with the help of UNION SELECT <Malicious code>. This gives the vulnerable query and we can able to add the malicious code in normal query with the help of union select. One can also use query to recover data from predetermined table. The consequences of these attack is dataset is returned from database is the union of the resultant of original query and malicious query.
Example: The example shows how hacker uses Union select statement in normal query and tries to get data.

SELECT accounts FROM users WHERE Login = ''UNION SELECT Pass from UserDetails WHERE account No=120420''

With the help of this example we can able to get the password from user details, so query is vulnerable and having the best way to extract data. Here hacker can get the information from user whose account no is 120420.UNION query is much vulnerable than tautology based attack. In may different application Pass from userDetails is shown alongside of the account information.

4) PiggyBacked Queries

Attack Intent: Executing remote command; select, extract and modifying data; performing DOS
Description: we can able to change or modify the intended query and tries to replace it with new vulnerable query in this attack. Hacker cannot alter the original query instead he uses new query to develop vulnerabilities called "PiggyBacked Queries ".In database there are two types of queries available, first the original intended queries and others are dangerous malicious queries as Piggybacked queries. Vulnerabilities to this type of attack is much more cause it having the change in data or modifying data.
Example: Hacker has many option to edit query like to drop certain important table or to add some new information detail of unauthorized person. Here Hacker is dropping table " '; drop table user- -" into pass field:

SELECT accounts FROM users WHERE
login= 'raj AND pass='';
drop table users -- ' AND pin=4444;

Here hacker is dropping the table whose login is raj and his pin is 4444 so the data of user raj is vanished. If that person is tries to login the account, he can unable to login the existing user. With the help of PiggyBacked Query user can able to do this kind of vulnerabilities [1].

B. DVWA Tool

There is constantly an approach to catch the thief if one can think like thief, this is also the same If anybody needs to recognize the attack then one must need to know that how attack can be happened. SQL Injection attack can be happened anywhere where database is available[5].

Moreover, the person should know Database Languages like MYSQL, Oracle, SQL Lite etc. The Normal SQL queries can get the data from the database, same as that of SQL Injection However for the bad purpose, Normal SQL queries can only get related information which is straightforward, where as SQLI queries can get the genuine data which are hidden and private[6].

The point of DVWA is to test various regular web vulnerability, with various difficultly levels, with a basic clear interface.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will illustrate how adding another layer of security to block certain malicious actions. There are also many public methods to bypassing these protections[7].

II. VULNERABILITIES

A. *OWASP Vulnerabilities*

DVWA is one of the most vulnerable tool in web application. OWASP top vulnerabilities are incorporated in DVWA.

In 2010,OWASPs top web application security risk:

- Insecure Cryptographic Storage
- Injection
- Cross-Site Scripting (XSS)
- Unvalidated Redirects and Forwards
- Insecure Direct Object References

Some of the web application vulnerabilities which DVWA contains;

1) *Insecure File Upload:* Enables a 'Hacker' to transfer malicious files on to the web server

2) *SQL Injection*: Allows a 'hacker' in which nefarious SQL statements are inserted into an entry field for execution.

3) *Easter eggs:* Full way Disclosure, verification bypass and some others

4) *Command Execution:* This performs orders on the hidden operating system.

5) *File Inclusion:* The vulnerability occurs due to the use of user-supplied input without proper validation.

6) *Cross Site Scripting (XSS):* A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy.

B. *DVWA Security*

The point of DVWA is to test various regular web vulnerability, with various difficultly levels, with a basic clear interface. There are two types of DVWA one is security level and other is PHP-IDS. In first security level section there are three levels as low, medium and high. Each section converts the condition of DVWA vulnerability. Naturally, the security level is set to High due to DVWA is stacked. Three different levels of Security in DVWA.

Low – This level contain no security i.e most vulnerable level.Programmer gives bad coding practice.

Medium – This security level is basically to a case for the user having awful coding practices, where the developer is attempt but neglect to secure an application. It is also used to test the skills of client to refine their vulnerable techniques.

High – This level is most secured level as the programmer is expert in coding and it uses vulnerable code to secured source code.

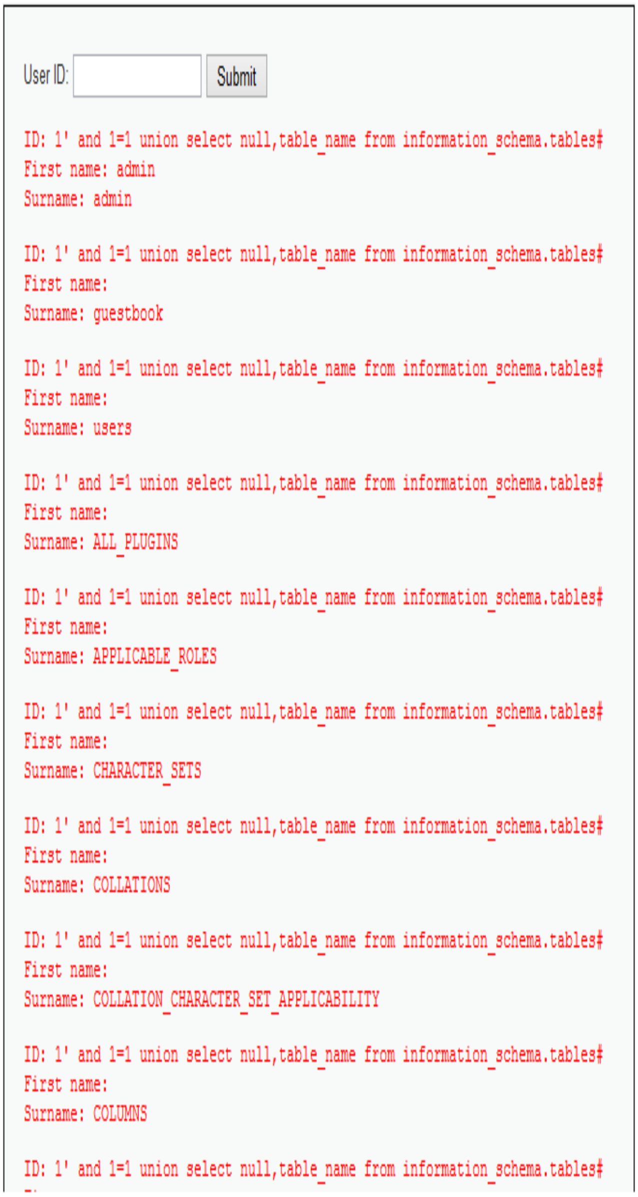
III. RESULT

In this paper work, DVWA Tool had SQL Injection Tab, figure 2 show user id and submit button. It contain 5 user Ids and their information. One should write a malicious code like

1' and 1=1# which gives first information of id and 1=1 gives that the query is true.

1' and 1=1 union select null,table_name from information_schema.tables# which gives information of first id and show all table name from the database().

Result will show in figure 2.



```
ID: 1' and 1=1 union select null,table_name from information_schema.tables#
First name: admin
Surname: admin

ID: 1' and 1=1 union select null,table_name from information_schema.tables#
First name:
Surname: guestbook

ID: 1' and 1=1 union select null,table_name from information_schema.tables#
First name:
Surname: users

ID: 1' and 1=1 union select null,table_name from information_schema.tables#
First name:
Surname: ALL_PLUGINS

ID: 1' and 1=1 union select null,table_name from information_schema.tables#
First name:
Surname: APPLICABLE_ROLES

ID: 1' and 1=1 union select null,table_name from information_schema.tables#
First name:
Surname: CHARACTER_SETS

ID: 1' and 1=1 union select null,table_name from information_schema.tables#
First name:
Surname: COLLATIONS

ID: 1' and 1=1 union select null,table_name from information_schema.tables#
First name:
Surname: COLLATION_CHARACTER_SET_APPLICABILITY

ID: 1' and 1=1 union select null,table_name from information_schema.tables#
First name:
Surname: COLUMNS

ID: 1' and 1=1 union select null,table_name from information_schema.tables#
```

Fig 2: Malicious Query in DVWA tool

Henceforth one can write a malicious code and try to collect malicious SQL Injection queries.

IV. Conclusion

DVWA can be used in a number of ways. By showing practical examples and setting challenges is used to teach security in web application for the students. It is used as just a learning tool, DVWA is planned all things considered to be as simple as conceivable to set up and utilize. There is numerous data inside DVWA to learn beginner. DVWA likewise utilized as a kind of perspective to secure coding, application against SQL Injection is secured if developer is not exactly beyond any doubt about it, So DVWA is one such tool to use to understand the SQL injection.

References

[1] Amir mohammad Sadeghian, Zamani Mazdak, Azizah Abd. Manaf, “SQL Injection Vulnerability General Patch Using Header Sanitization”, 2014 International Conference Computer, Communication and Control Technology.

[2] Joshi Anamika, V. Geetha, “SQL Injection Detection using Machine Learning”, International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT 2014).

[3] Buja Geogiana, Dr. Kamarularifin Bin Abd Jalil, Dr. Fakariah Bt. Hj Mohd Ali, Teh Faradilla Abdul Rahman, “Detection Model for SQL Injection Attack: An Approach for Preventing a Web Application from the SQL Injection Attack” 2014 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE) , April 7 - 8, 2014, Penang, Malaysia.

[4] Lwin Khin Shar and Hee Beng Kuan Tan, “Defeating SQL Injection”,2013 Published by the IEEE Computer Society.

[5] http://www.dvwa.co.uk/

[6] http://www.dvwa.co.uk/forum

[7] http://dvwa.svn.sourceforge.net/svnroot/dvwa

[8] Djuric Zoran, “A Black-box Testing Tool for Detecting SQL Injection Vulnerabilities” 2013 Informatics and Applications (ICIA),2013 Second International Conference.

[9] Komiya Ryohei, Paik Incheon, Masayuki Hisada,” Classification of Malicious Web Code by Machine Learning” 2011 Awareness Science and Tec bhnology (iCAST), 2011 3rd International Conference.

[10] Gupta Mukesh Kumar, Govil Mahesh Chand, Singh Girdhari, “An Approach to Minimize False Positive in SQLI Vulnerabilities Detection Techniques through Data Mining”,2014 Signal Propagation and Computer Technology (ICSPCT), International Conference.