Detection of SQL Injection Attack and Various Prevention Strategies

Priyanka, Vijay Kumar Bohat

Abstract---The internet is a demanding technology which is working its way into all aspects of our civilization. So security is the main critical part in our daily life. The requirements of information security and website security within an organization have undergone several changes in the last several decades. Security is a broad topic and covers a multitude of sins. This paper is written with the basic programmer and information security expert, explaining the concepts which are needed to read through the hype in the market place and understand the risks and how to deal with them. We go on to consider risk management, network threats, firewalls, protection from SQL Injection.

SQL Injection is a web attack mechanisms which is being used by hackers to misuse the data of that website. It is hoped that this paper will help the reader to provide a wider perspective on security and better understand how to handle and manage risk related to security issues of website personally at client–end and at server-end.

Index Terms--- SQL Injection; SQL Injection prevention; SQL Injection detection; website security.

I. INTRODUCTION

In this 21st century, where all the important works of day-to day life are being done on internet, collecting any new information we search it from Google, then for anything which we want to buy we can do online shopping from e-commerce websites, for uploading assignment by faculty and students download it using internet only, and many more other work done on internet using various websites. SQL Injection is a web attack mechanisms which is being used by hackers to misuse the data of that website. SQL Injection is one among the many web attacks used today and is being applied on several websites which are not secured properly. In this type of web attack the hacker takes advantage of wrong and incomplete coding of a website which allows him to inject SQL Injection codes into admin login site and the he gains the access to the data present within that website's database.

In short, SQL Injection attack occurs due to vulnerabilities present in the website that allows the hacker to bye passes the SQL statements and hence enters into the database queries directly.

Manuscript received April 2013.

Priyanka, Pursuing M.Tech in Computer Science and Engineering from Lovely Professional University, Punjab, India

Vijay Kumar Bohat, Computer Science and Engineering from Lovely Professional University, Punjab, India

II. HOW SQL INJECTION WORKS

A. Finding admin login page using Google Dorks:

- "inurl:admin.asp"
- "inurl:login/admin.asp"
- "inurl:admin/login.asp"
- "inurl:adminlogin.asp"
- "inurl:adminhome.asp"
- "inurl:admin_login.asp"
- "inurl:administratorlogin.asp"
- "inurl:login/administrator.asp"
- "inurl:administrator_login.asp"

B. SQL Injection Attack

Un-authorized Access Attempt by putting vulnerable code in the input field as:
password→ 'or''='

C. SQL statement becomes

select count(*) from users where username = 'priyanka' and password = 'or''='
Checks that if password is empty OR 0=0 ,1=1 which is always true, permitting access.

III. CLASSIFICATION OF SQL INJECTION

The SQL Injection attack can be categorized as:

A. String based SQL Injection:

SQL Injection is the first step in entry to exploiting or hacking any website available on internet. String Based SQL Injection is an attack in which the attacker inputs the code, which is a malicious code, in the login page of the admin, with which he is able to gain the access of that website on which this attack is successful.

The login page of the admin that takes usernames and password works using the SQL Query as [9] [5]:
SELECT * FROM table_name WHERE username= 'priyanka' AND password= 'password';
Now the problem which can occur in this SQL query statement is that two input variables, i.e. username and password are inputting the values directly without making check filter on these values. So the attacker can now bypass the authentication of the admin by making use of malicious code as [1] [9] [8]:
Username= **'or''='**
Password= **'or''='**

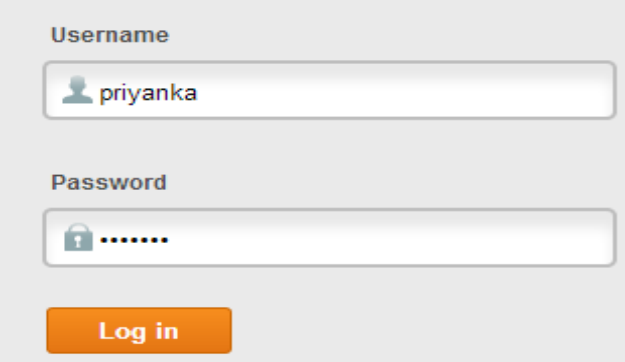

Figure 1

Using this malicious string based code the above written SQL Query becomes like this:
SELECT * FROM table_name WHERE username='priyanka' AND password= '1'= '1' OR '0' = '0'
The admin login page for attack can be found using Google dorks [9]:
inurl:admin
inurl:adminlogin
inurl:admin_login

B. Error based SQL Injection:

While tackling this type of attack the point which must be known is:
"Data is in the columns and the columns are in tables and the tables are in the database."
Using Error based SQL Injection:

a. Finding SQL Injection vulnerable sites.
b. Find number of tables.
c. Select and find vulnerable table
d. Get the MySQL Version.
e. Get the Database Name.
f. Get the Table Names.
g. Get the Column Names.

The password is in encrypted form which can be decrypted using Hash Decryptors

C. BLIND SQL INJECTION

Blind SQL injection is a type of SQL Injection attack in which the attacker tries attack on the database by entering true or false questions and then on the basis of website's application response it determines the answer. As per OSWAP description, this attack is being used when any given web application is configured to show generic error messages, but has not mitigated the code that is vulnerable to SQL injection[2].
In blind SQL Injection there is error message shown on the website when the attacker tries to star the attack from the database with SQL Query's syntax error. Blind SQL injection is mostly similar to normal SQL Injection, the only difference is the method with which the attack is performed on the data of the given website..[2]

D. SQL INJECTION WITH SHELL UPLOADING

A shell behaves similar to the software which providesa user with an interfaceof an operating system that provides access to the services of a kernel. After finding browse button on a website admin login area, the attacker tries to upload the shell on that browse button and which can attack that website and the attacker can get full access of that website under his control or even its server also.

IV. ANALYSIS

Here we are trying to show the analysis done by various organizations in year 2010 to 2013 on the basis of number of attacks present these days and the ranking of SQL Injection among them.[5][6]

A. According to Data Breach Investigation Report-2012 by Verizon RISK Team, top threat actions used against larger enterprises is as following[5]:

Rank	Overall Rank	Variety	Category	Breaches	Records
1	3	Use of stolen login credentials	Hacking	30%	84%
2	6	Backdoor (allows remote access/control)	Malware	18%	51%
3	7	Exploitation of backdoor or command and control channel	Hacking	17%	51%
4	9	Tampering	Physical	17%	<1%
5	1	Keylogger/Form-grabber/Spyware (capture data from user activity)	Malware	13%	36%
6	11	Pretexting (classic social engineering)	Social	12%	<1%
7	5	Brute force and dictionary attacks	Hacking	8%	<1%
8	15	SQL injection	Hacking	8%	1%
9	20	Phishing (or any type of "ishing)	Social	8%	38%
10	22	Command and control (listens for and executes commands)	Malware	8%	36%

Figure 2Top ten threats used against larger enterprises

B. As per Imperva Hacker Intelligence Initiative, Monthly Trend Report #13, it analyzed that SQL Injection and DDOS are most popular topics these days[6].

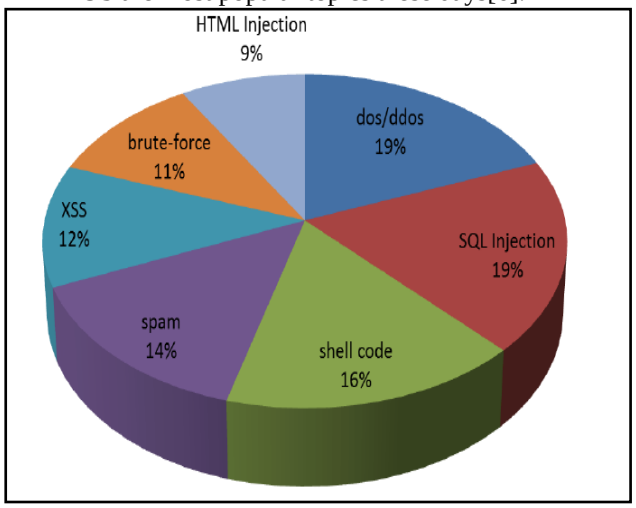

Figure 3 Percantage of threats September 2011-September 2012

C. As per Imperva Hacker Intelligence Initiative, Monthly Trend Report #13, top 7 attacks these days with ranking are[6]:

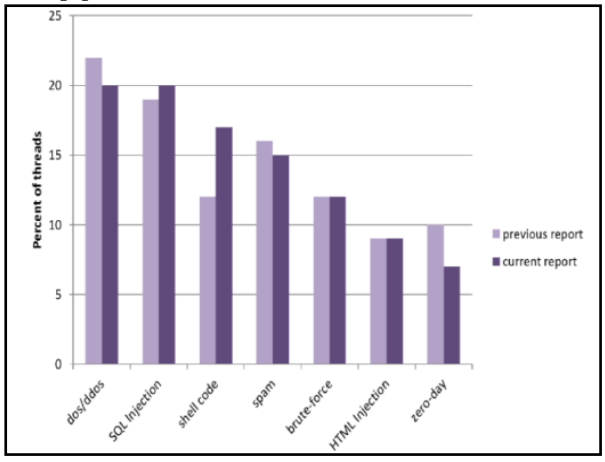

Figure 4Top 7 attacks changes in conversation

D. Imperva Application Defence Center compared out top ten security threads for year 2010 and 2013 as given in the figure below. This indicates that ranking of SQL Injection comes out be on number 3rd in 2013 from number 5th in the year 2010[6].

Ranking	2013 Top Threats		2010 Top Threats
1	Excessive and Unused Privileges		Excessive Privilege Abuse
2	Privilege Abuse		Legitimate Privilege Abuse
3	SQL Injection	↑	Privilege Elevation
4	Malware	NEW	Exploitation of Vulnerable, Miconfigured Databases
5	Weak Audit Trail	↑	SQL Injection
6	Storage Media Exposure	↑	Weak Audit Trail
7	Exploitation of Vulnerabilities and Misconfigured Databases	↓	Denial of Service
8	Unmanaged Sensitive Data	↑	Database Communication Protocol Vulnerabilities
9	Denial of Service	↓	Unauthorized Copies of Sensitive Data
10	Limited Security Expertise and Education	NEW	Backup Data Exposure

Figure 5Top 10 threats in year 2010 v/s 2013

V. PRESENT WORK

A. Problem Formulation

We need to scan a website at front-end after scanning the website we need to find vulnerability. For particular vulnerability we need to secure them. For example if the website has database connectivity at backend then we want to protect that website from various types of attacks like SQL Injection.

B. Proposed Architecture

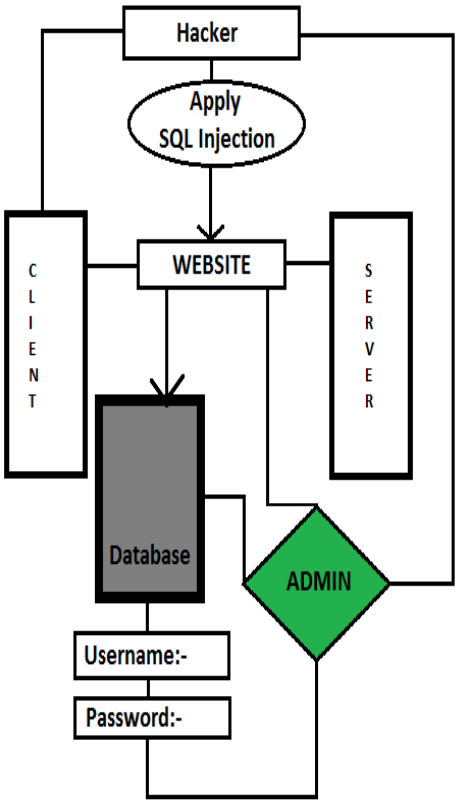

Figure 6Proposed Architecture for interaction between Designer and Admin of a Website

C. Objective of Study

"To provide security at client side and server side."

The list of attacks includes following and many more:

S No.	Attacks	Platforms
1	SQL Injection	Php, asp
2	Word press-CMS	Php
3	XSS	Php, asp
4	LFI/RFI	Php, asp
5	CSRF	Php, asp
6	DNN	Asp
7	SSI	Asp
8	Symlinking	Php, asp
9	DOS	Windows/Linux
10	DDOS	Windows/Linux
11	Brute Force	Windows/Linux
12	Buffer overflow	Windows/ Linux

Security of website: Security is an important issue for recent generation Web sites.

Security issues can be divided into two categories: *security of a system* in which we ensure that other people cannot change your Web site and *security of information* in which we ensure that the customer details from an online store are safe.

D. Research Methodology

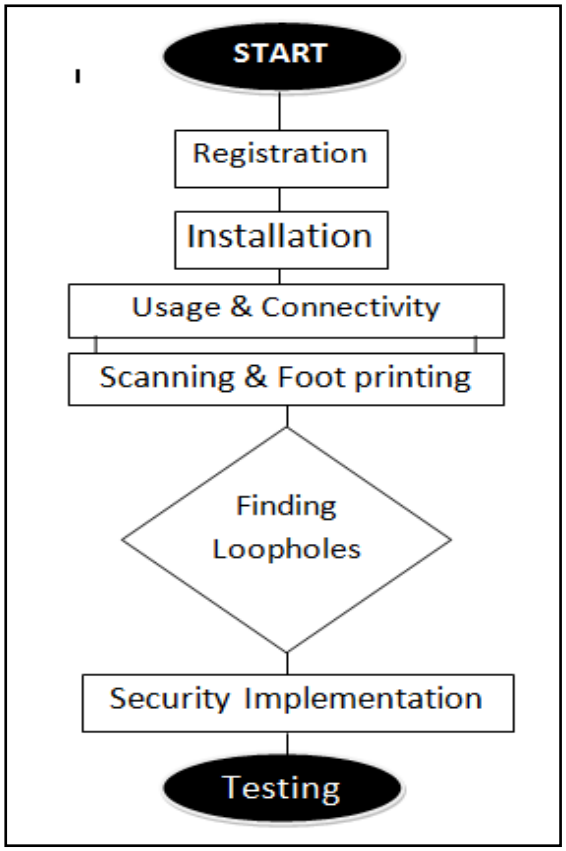

Figure 7 Research Methodology Steps

VI. IMPLEMENTATION

Security of a website can be done using four security methods:

A. *Using function mysql_real_escape_string*

Always use mysql_real_escape_string () function before sending the variable to the SQL query
For example
$username=mysql_real_escape_string($_POST['username']);
$password=mysql_real_escape_string($_POST['password']);
If an intruder inject ‘ OR 1 in the user name and password field then the value of the $username and $password will become \’ OR 1 which is not going to harm us anymore.

B. *Autocomplete Enabled*

Autocomplete is a HTML tag attribute that is used to disable theform ofthe browser for auto completion.
Use a tag of autocomplete in the code as:
<form name="form1" method="post" action="checklogin.php" autocomplete="off">

C. *Using .htaccess security*

Htaccess is an abbreviation for Hypertext Access. It is a configuration file that controls the directories and the subdirectories contained in it.The .htaccess file helps lot of control and lets you easily redirect the pages, password protect directories and many things more.
In .htaccess file write the following code for security:
options –indexes

D. *Add-on Security Points*

Protecting a website or web server is possible only by continued efforts.

- Use Open Source Scripts
- Update your website Constantly
- Use Strong and encrypted Passwords
- Secure Admin Email Address
- Password protect the Database
- Delete the Folder where installation was done
- Change privileges according to each users like, File & Folder Permissions
- Use Secured FTP Access
- Restrict Root Access
- Ensure that .htaccess file is present in your website
- Use security plugins

VII. EXPERIMENTAL RESULTS

While working on the demonstrated website we used various security implementation steps as described above in the implementation section. With these security methods we are able to reduce the total number of alerts from four to zero alert by scanning the website using vulnerability tool. The proposed work gives the following results:

A. *Demo Website BEFORE SECURITY*

Implementation:

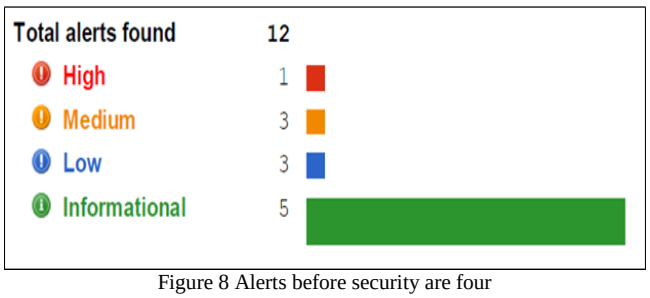

Figure 8 Alerts before security are four

B. *Demo Website AFTER SECURITY*

Implementation:

The results on website scanning after applying security implementation decreases the total number of alerts from four to zero now.

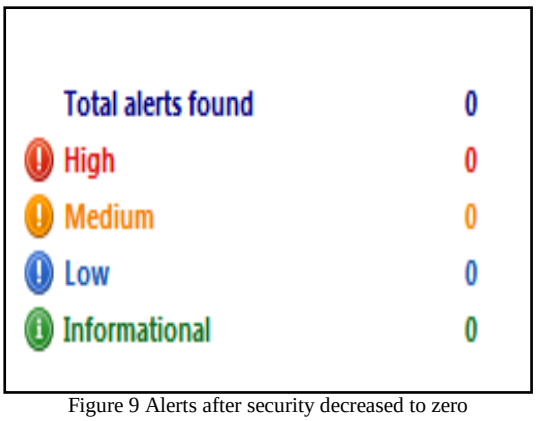

Figure 9 Alerts after security decreased to zero

VIII. CONCLUSION AND FUTURE WORK

In this paper, we have tried to explain how to detect and prevent the SQL Injection attack on the websites. There are various techniques used to secure SQL Injection attack named as , by .htaccess configuration file, by php function , mysql_real_escape_string and by disabling autocomplete form.

But lots of websites are present on the internet, and it is obvious that black-hat hackers are always busy in finding the new techniques and applying the attack in a new way. So new countermeasures are required to deal with them. We believe that the methods given in this paper to define SQL Injection and its prevention techniques will be beneficial for security of a website. But due to black hacker’s strategy, further investigation in this domain will be required and we will try to survey continuously for this domain and find the new protection measures.

REFERENCE

[1] A Tajpour, A., Masrom, M., Heydari, M.Z., and Ibrahim, S., SQL injection detection and prevention tools assessment. Proc. 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT’10) 9-11 July (2010), 518-522
[2] Khaleel Ahmad, Jayant Shekharand, K.P. Yadav(2010),” Classification of SQL Injection Attacks” VSRD technical and non-technical journal
[3] Shubham Shrivastava, Rajeev Ranjan Kumar Tripathi, Attacks Due to SQL injection & their Prevention Method for Web-Application, International Journal of Computer Sciecne and information technologies, Vol 3 (2), pp.3615-3618, 2012.
[4] Chad Dougherty (2012) “Practical Identification of SQL Injection Vulnerabilities” United States Computer Emergency Readiness Team (US-CERT October 25, 2012)
[5] Parveen Kumar (2013) “The Multi-Tier Architecture for Developing Secure Website with Detection and Prevention of SQL-Injection Attacks” International Journal of Computer Applications (0975 – 8887)Volume 62– No.9, January 2013
[6] https://www.owasp.org/index.php/SQL_Injection
[7] http://www.imperva.com/docs/hii_monitoring_hacker_forums_2012.pdf
[8] http://www.acunetix.com/websitesecurity/sql-injection/
[9] http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2012-ebk_en_xg.pdf
[10] http://www.imperva.com/docs/hii_monitoring_hacker_forums_2012.pdf