PREVENTING SQL INJECTION

By

Sumana Sattar
(2003-NUST-BIT-45)

A project report submitted in partial fulfillment of
the requirement for the degree of
Bachelors in Information Technology

In

NUST Institute of Information Technology
National University of Sciences and Technology
Rawalpindi, Pakistan
(2007)

CERTIFICATE

It is certified that the contents and form of thesis entitled **"Preventing SQL Injection"** submitted by Ms. Sumana Sattar has been found satisfactory for the requirements of the degree.

Project Advisor: __________________
Lecturer (Mr. Sheharzad Khattak)

Co-Advisor: ______________________________
Lecturer (Mr. Fuzan Mirza)

Member: ______________________________
Lecturer (Mr. Abdul Ghafoor)

Member: ______________________________
Lecturer (Mam Sana Khalique)

DEDICATION

In the name of Allah, the Most Beneficent, the Most Merciful. I would like to dedicate my work to my parents.

ACKNOWLEDGEMENTS

All praise to Allah Almighty Who has bestowed upon me a lot of His blessings and made me capable of doing things which I never even thought of doing. I am deeply beholden to my supervisor Sir Shahrzad Khattak for his continuous assistance, encouragement, suggestions and guidance and without his valuable implications I would not be able to complete this project. I also thank my co-advisor Fauzan Mirza and committee members (Mam Sana and Sir Abdul Ghafoor) for being supportive. I would like to thank all the teachers who have taught me throughout these four years of course. I am thankful to NIIT Administration for helping me in administrative problems. In the end I would like to thank my parents who have always encouraged me.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

SQL injections are hacking techniques through which hackers gain unauthorized access to the database. It has been declare as one of the most serious threat to web-base application. Not only the web-base application but all the applications which have dynamically generated SQL queries are vulnerable to SQL injections. SQL injections are of many types it all depend upon the creativity of the hacker how he wants to attack the database. But mainly it has been classified as four types: Select, Update, Authorization bypass and Insert. Lot of research has been carried out and many solutions have been proposed but all the solutions have some limitations. The best solution ever proposed is parse tree technique. Parse Tree technique has been implemented in this project. The basic concept behind the Parse Tree is that it dynamically generates two parse trees from SQL query. One with user input and other without user input And compare at run time If they are equal it will be consider as safe query and it will be allowed to access the database. Otherwise it will be consider as an SQL injection attack and It will not further be proceed.

Chapter 1

INTRODUCTION

SQL injection technique are use to exploit the web base application not only web base application all the application that has a dynamic database at there back end are vulnerable to such attacks. SQL injections are very easy to learn and master. SQL injection is a hacking technique that allows hacker to enter the database. The hacker can access, update and delete the records from the database. Hackers test for SQL injection vulnerabilities by sending the application an invalid input that would cause the server to generate an error message. Based on these error messages the hacker will learn about the back end database and will make more attempt to learn more about it. He will continue doing this until he has enough information to attack the database. SQL injections are of many types. It all depends on the mind of the hacker how creative he is. Mainly it is classified into four types: Insert, select, update and authorization bypass.

Lot of research has been carried out and many solutions have been proposed to prevent SQL injections but all the solutions have some limitations. The best solution ever proposed is parse tree technique. In this project Parse Tree technique has been used to prevent SQL injections. The basic concept behind the Parse Tree is that it dynamically generates two parse trees from SQL query. One with user input and other without user input And compare at run time If they are equal it will be consider as safe query and it will be allowed to access the database. Other wise it will be consider as an SQL injection attack and it will not further be proceed.

1.1 MOTIVATION

SQL injections are very easy to learn and master. Lot of information related to this issue is provided over the net and any one with little knowledge of SQL queries and programming skills will be able to hack the websites. And once the hacker is able to enter the database, he can modify, add, delete and access the contents of the database .Most of the Pakistani developers are unaware of SQL injections. Therefore most of the Pakistani websites are vulnerable to SQL injections. This is a very serious issue, which needed to be solved. Lot of research has been carried out and lot of solution has been proposed but with some limitations. Parse tree is the solution with little limitation which has been implemented in this project.

1.2 PROBLEM STATEMENT

"Problem is to develop a system which prevent the SQL injection attack in most efficient manner"

1.3 PROJECT SCOPE

My intention is to learn about SQL injections and use them to hack websites. After that I shall attempt to provide web security against those attacks by using Parse tree technique.

1.4 PROJECT DOMAIN

Following are the method by which database can be hacked

1. Direct: In this method hacker gain direct access to desktop application. Here a hacker is within the company or outsider who got the chance to sit on the system.

2. Indirect: In this method user could be anyone, anywhere he tries to access database of the website by sitting anywhere in this world.

3. Platform: This is a generic attack e-g worms.

In this project I am trying to prevent indirect method i-e I m trying to secure web base application from SQL injection that are using Oracle 9i at the back end.

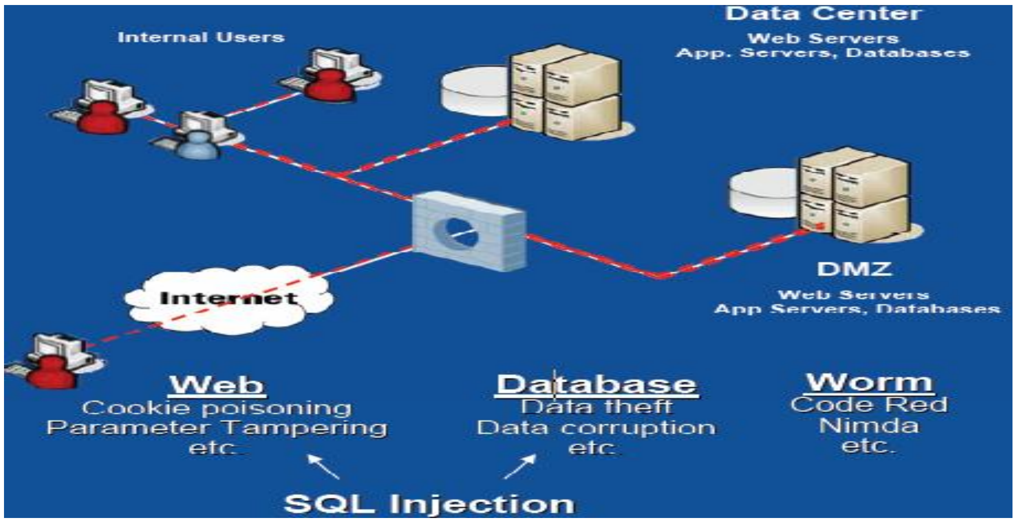


Figure -1.1 Database Threats and Vulnerabilities

1.5 SOFTWARE TOOLS

Following are the tools and software used for my project

1. .NET

2. Oracle 9i

3. Web server

4. Database server

1.6 DELIVERABLES

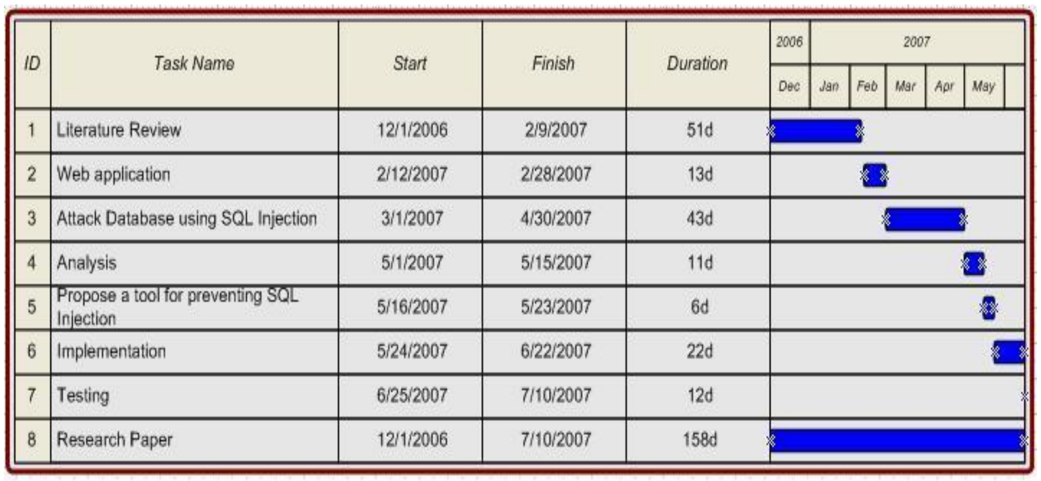
1 Web application

2 Attacks on database

3 Tool for preventing Attacks

4 Research paper

1.7 TIME LINE



ID	Task Name	Start	Finish	Duration
1	Literature Review	12/1/2006	2/9/2007	51d
2	Web application	2/12/2007	2/28/2007	13d
3	Attack Database using SQL Injection	3/1/2007	4/30/2007	43d
4	Analysis	5/1/2007	5/15/2007	11d
5	Propose a tool for preventing SQL Injection	5/16/2007	5/23/2007	6d
6	Implementation	5/24/2007	6/22/2007	22d
7	Testing	6/25/2007	7/10/2007	12d
8	Research Paper	12/1/2006	7/10/2007	158d

Figure-1.2 Time lines

1.8 REPORT STRUCTURE

This document will throw light on all the aspects of the project, the technique being implemented, the existing techniques and the details of the developed tool. This document has been divided in the following ways.

Chapter 2: Literature Review The second chapter will throw light on what are SQL injections and how they can be prevented.

Chapter 3: Methodology: This chapter will highlight the architecture of the project and will throw lights on the different modules that make up the tool.

Chapter 4: Implementation: In fourth chapter the detailed working of the proposed technique will be discussed. This chapter will focus on the concepts related to software engineering and the implementation details of the proposed solution.

Chapter 5: Testing. This chapter will highlight the techniques that were used in order to test the proposed approach and its efficiency. This chapter explains the testing of the implemented technique.

Chapter 6: Conclusion and References. This chapter summarizes the whole document.

Chapter 2

LITERATURE REVIEW

2.1 WHAT IS SQL INJECTION?

SQL injections are hacking techniques that are used by hacker to gain unauthorized access to the database. The hacker is then free to extract, modify, add, or delete records from the database. [1]

Hacker first tests the desire application for the vulnerabilities by giving invalid input which in return will cause the server to generate an error message. Though this error message the attacker will learn about the back end data. And will make more attempts to learn more and more about the database. When he had enough knowledge to attack the website he will try an SQL injection on the website. SQL injections are of many types. It all depends on the mind of the hacker how creative he is. Mainly it is classified into four types: Insert, select, store procedure and authorization bypass.

SQL injections are very easy to learn and master. Lot of information related to this issue is provided over the net and any one with little knowledge of SQL queries and programming skills will be able to hack the websites. And once the hacker is able to enter the database, he can modify, add, delete and access the contents of the database .Most of the Pakistani developers are unaware of SQL injections. Therefore most of the Pakistani websites are vulnerable to SQL injections. This is a very serious issue, which needed to be solved. Lot of research has been carried out and lot of solution has been proposed but with some limitations. Parse tree is the solution with little limitation which has been implemented in this project.[8]

2.2 WHAT'S VULNERABLE

SQL injections can be made possible only because of the dynamic queries. Because it is very difficult to validate the dynamically generated queries and hacker take advantage of this and is able to view, modify the back end data. Now a day it is becoming a very common practice. Because lot of information related to this issue is present over the net.[2]

2.3 STATISTIC OF HACKING

Following are the statistics of hacking in year 2006 [3]

1. Cross Site Scripting (21.5%)
2. SQL Injection (14%)
3. PHP includes (9.5%)

4. Buffer overflows (7.9%)

2.6 TYPES OF SQL INJECTION

Following are the type of SQL Injections [4]

- Authorization bypass
- INSERT
- Store Procedure
- SELECT

2.6.1 Authorization Bypass:

The simplest SQL injection technique is bypassing form-based logins. Let's say that the web application's code is like this:

"SELECT Username FROM Users WHERE Username = ' " +Username + " ' AND Password = ' " + Password + " ' "

Enter following values in the text box

Login: ' OR ''='

Password: ' OR ''='

This will give SQLQuery the following value:

SELECT Username FROM Users WHERE Username = '' OR ''='' AND Password = '' OR ''=''

This will make the statement true because empty is always equal to empty. And in this way the user will by pass the login page.

2.6.2 Insert

Normally an insert statement looks like this:

Insert into My_table Values (‘myvalue1’, ‘myvalue2’, ‘myvalue3’);

instead of giving user name, email and phone no u enter the following values [7]

Name: ‘ + (SELECT TOP 1 value_fist from Mytable) + ‘

Email:my.email@whtever.com

Phone:12345566

The resulting SQL statement will be like this

INSERT INTO My_table values (‘ “ ‘ + (SELECT TOP 1 value_fist from Mytable) +

’ ‘ , ‘my.email@whtever.com’, ‘12345566 ’)

The result of this query will be the fist field from the table my table.

2.6.3 Using Stored Procedures

Stored procedure were known for the prevention of SQL injection but now it is been detected that stored procedures are still vulnerable to SQL injections. Lets take an example.[5]

```
CREATE PROCEDURE dbo.table

        @my_name sysname

AS
```

EXEC (‘CREATE TABLE ‘+@my_Name+’(column_1 varchar(20),column_2 varchar (20))’);

GO

By looking at the above code it is obvious that what ever is entered in the ‘my_name’ will be appended in the create table. So instead of giving a valid input u give the input given below.

‘; SHUTDOWN WITH NOWAIT;--

What actually happens it will first close the brackets of the create table then ‘;’ will end the create statement. And ‘SHUTDOWN WITH NOWAIT’ will stop the SQL server without waiting for any other request.

2.6.4 Select

Select statements are also vulnerable to SQL injections. This can be proved by the following example.

my_query = “SELECT mycol1, mycol2, mycol3 WHERE ISBN = ‘ ” +user_input+ “ ‘ ”

if we inject the following query

‘UNION ALL SELECT my_desire col WHERE ‘ ’ = ‘ ‘

The resultant query will be

SELECT mycol1, mycol2, mycol3 WHERE ISBN = ‘ ‘‘UNION ALL SELECT

my_desire col WHERE ‘ ’ = ‘ ‘;

so will be able to view the my_deiser_col values.

2.7 EXAMPLES OF HACKED WEBSITES

Following are the Pakistani websites which were found vulnerable to SQL injections.

1. The website shown in the figure 3 claim to be largest web portal and they are vulnerable to SQL injections**.**

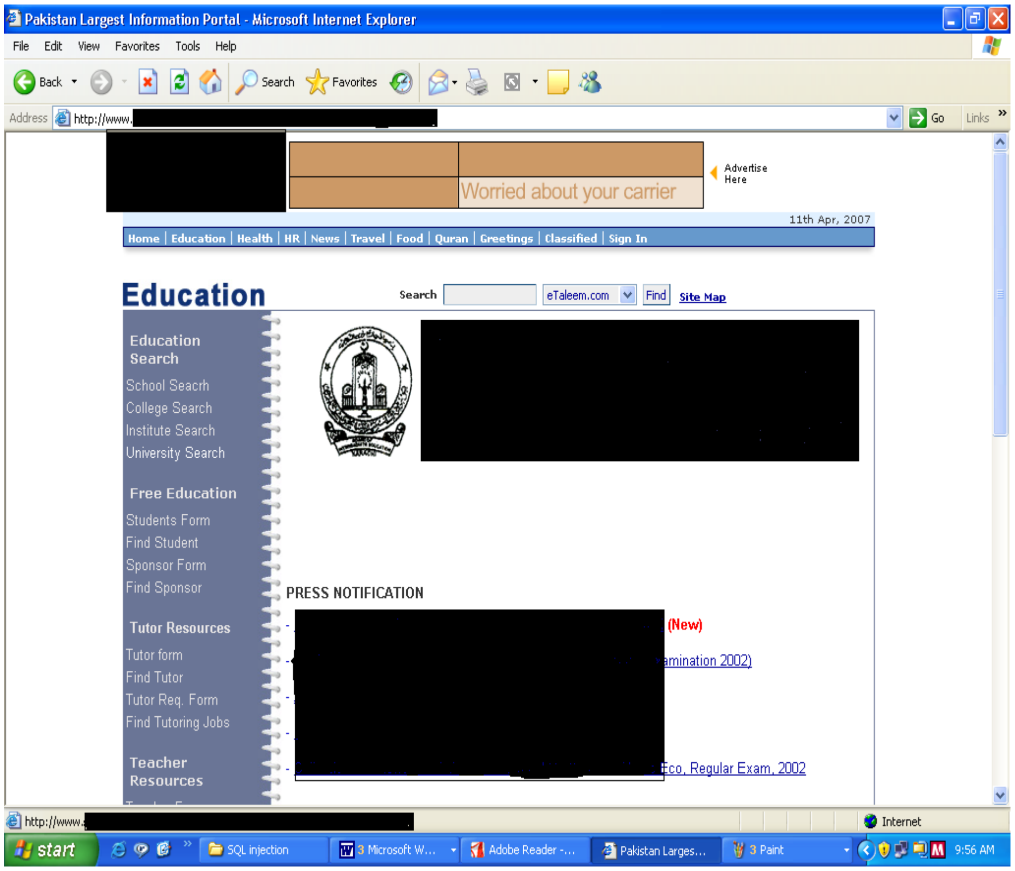
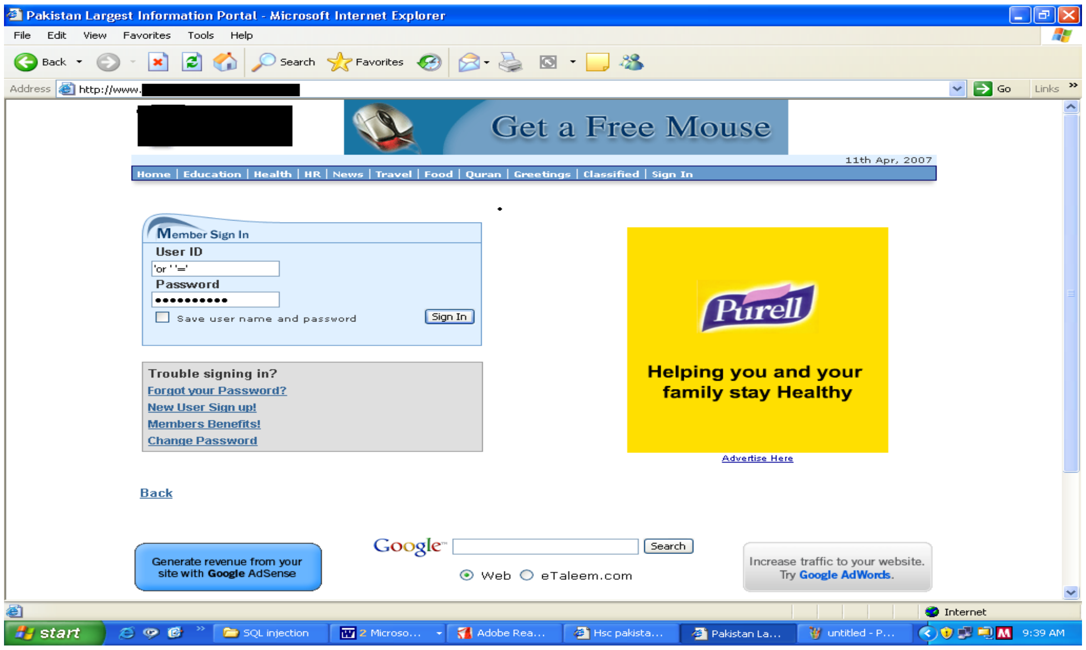


Figure-2.1 Main page of Pakistan largest search portal

If we enter the SQL injection in the login name and password field it be allowed to go further and will be able to login as Adnank which is probably the first record in the data base. Figure 4 shows the login page of Pakistan largest search portal. And figure 5 shows the home page of Adnank.

]


Pakistan Largest Information Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.[redacted]
Go Links
[redacted]
Get a Free Mouse
11th Apr, 2007
Home | Education | Health | HR | News | Travel | Food | Quran | Greetings | Classified | Sign In
Member Sign In
Trouble signing in?
Forgot your Password?
New User Sign up!
Members Benefits!
Change Password
Purell
Helping you and your family stay Healthy
Advertise Here
Back
Generate revenue from your site with Google AdSense
Google Search
Web eTaleem.com
Increase traffic to your website. Try Google AdWords.
Internet
start SQL injection 2 Microso... Adobe Rea... Hsc pakista... Pakistan La... untitled - P... 9:39 AM

User ID: 'or ''='

Password: ●●●●●●●●●●

☐ Save user name and password

Sign In

Figure-2.2 login page of Pakistan largest search portal

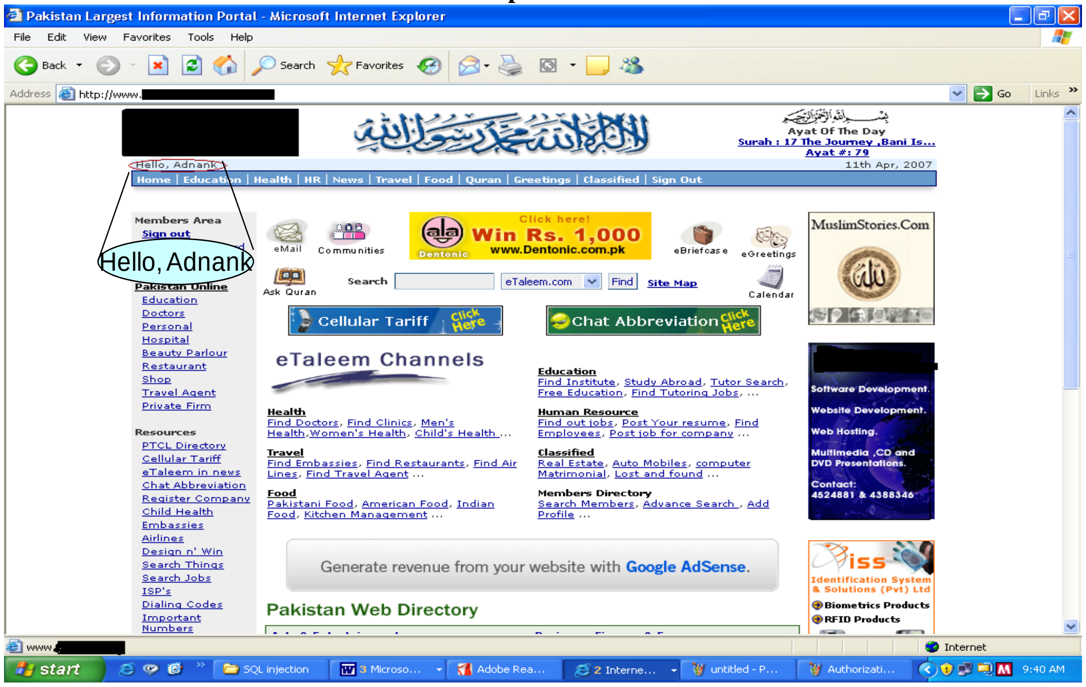

Pakistan Largest Information Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.[redacted]
Go Links
[redacted]
Ayat Of The Day
Surah : 17 The Journey ,Bani Is...
Ayat #: 79
Hello, Adnank
11th Apr, 2007
Home | Education | Health | HR | News | Travel | Food | Quran | Greetings | Classified | Sign Out
Members Area
Sign out
Hello, Adnank
Pakistan Online
Education
Doctors
Personal
Hospital
Beauty Parlour
Restaurant
Shop
Travel Agent
Private Firm
Resources
PTCL Directory
Cellular Tariff
eTaleem in news
Chat Abbreviation
Register Company
Child Health
Embassies
Airlines
Design n' Win
Search Things
Search Jobs
ISP's
Dialing Codes
Important Numbers
eMail
Communities
Click here! Win Rs. 1,000 www.Dentonic.com.pk
eBriefcase
eGreetings
Ask Quran
Search eTaleem.com Find Site Map
Calendar
Cellular Tariff Click Here
Chat Abbreviation Click Here
eTaleem Channels
Education
Find Institute, Study Abroad, Tutor Search, Free Education, Find Tutoring Jobs, ...
Health
Find Doctors, Find Clinics, Men's Health,Women's Health, Child's Health ...
Human Resource
Find out jobs, Post Your resume, Find Employees, Post job for company ...
Travel
Find Embassies, Find Restaurants, Find Air Lines, Find Travel Agent ...
Classified
Real Estate, Auto Mobiles, computer Matrimonial, Lost and found ...
Food
Pakistani Food, American Food, Indian Food, Kitchen Management ...
Members Directory
Search Members, Advance Search , Add Profile ...
Generate revenue from your website with Google AdSense.
Pakistan Web Directory
MuslimStories.Com
[redacted]
Software Development.
Website Development.
Web Hosting.
Multimedia ,CD and DVD Presentations.
Contact: 4524881 & 4388346
iss
Identification System & Solutions (Pvt) Ltd
Biometrics Products
RFID Products
www.[redacted]
Internet
start SQL injection 3 Microso... Adobe Rea... 2 Interne... untitled - P... Authorizati... 9:40 AM

Figure-2.3 home page of Adnank

2. Figure 6 shows another web site which is vulnerable to SQL injections. Here students can view there results of FSc and matric.

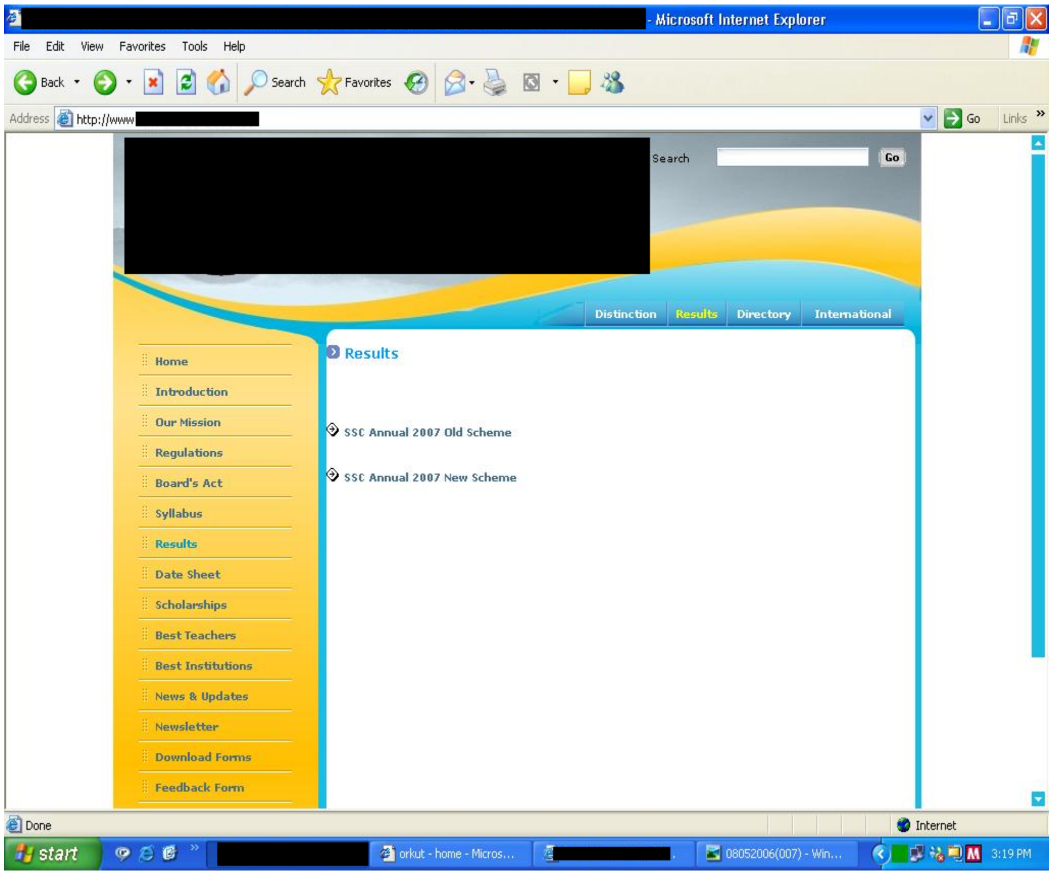
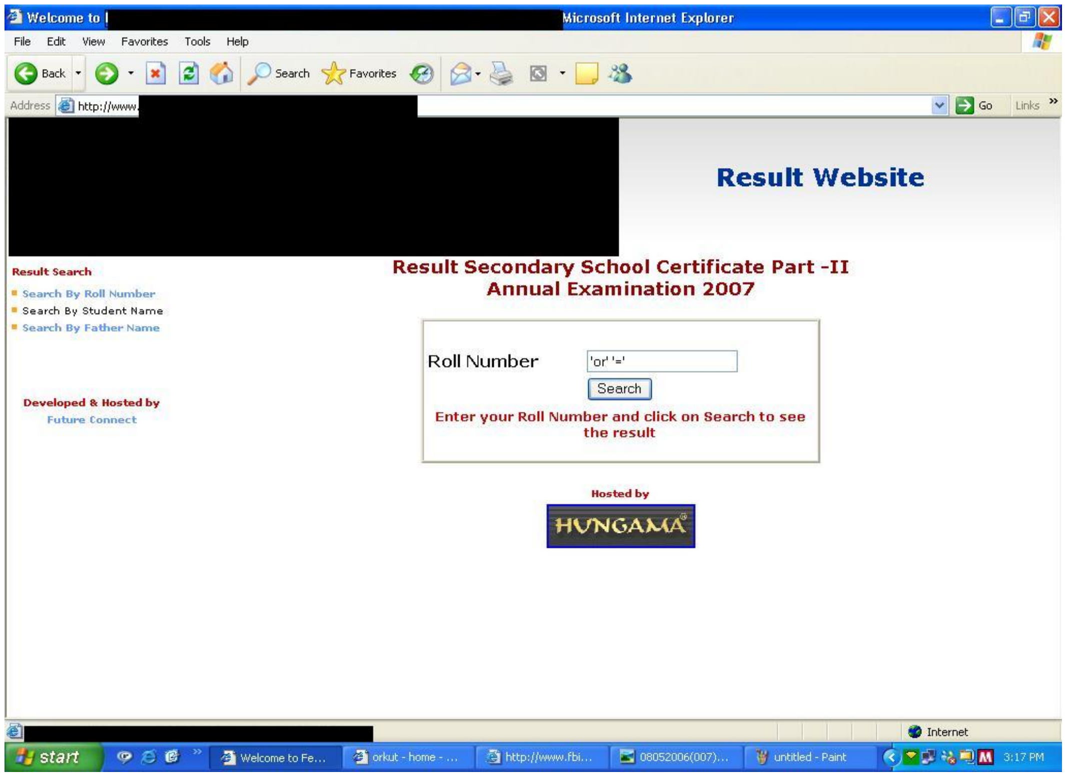


Figure-2.4 Main page

Figure 7 shows the page where students can enter the role number to view the result. Instead of giving a roll SQL injection was entered and I was able to view the result of Waqas Hussain as shown in the Figure 8.


Welcome to [redacted] Microsoft Internet Explorer

Address http://www.[redacted]

[redacted]

Result Website

Result Search
- Search By Roll Number
- Search By Student Name
- Search By Father Name

Developed & Hosted by
Future Connect

Result Secondary School Certificate Part -II
Annual Examination 2007

Roll Number 'or' '='

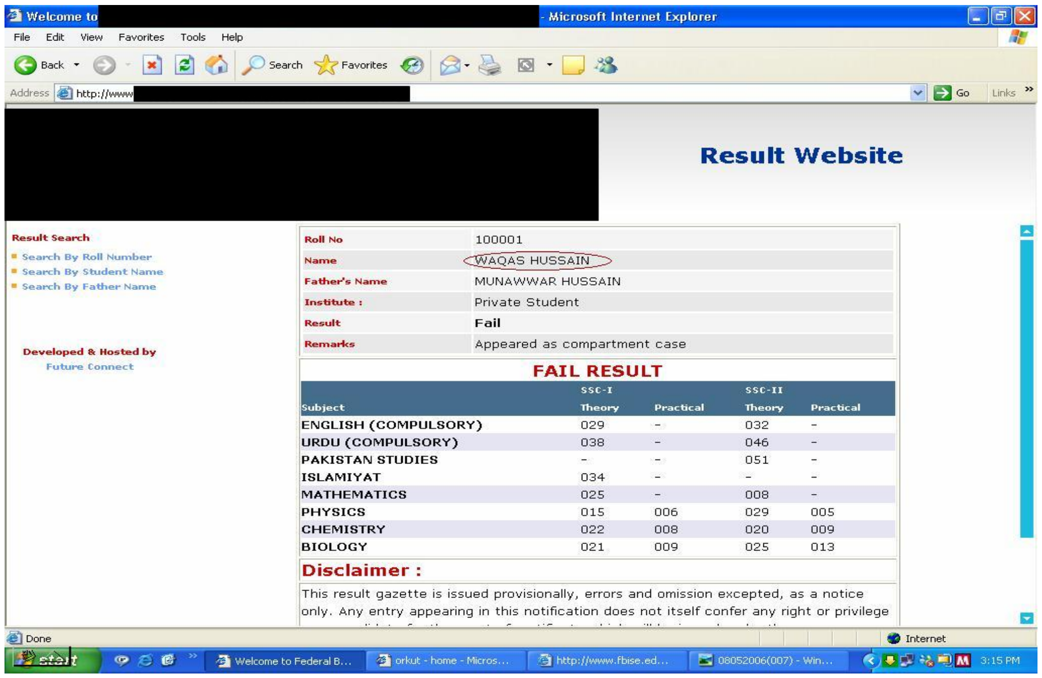
Search

Enter your Roll Number and click on Search to see the result

Hosted by
HUNGAMA

[redacted]

Figure-2.5 Page to enter roll no to view the result


Welcome to [redacted] - Microsoft Internet Explorer

Address http://www[redacted]

[redacted]

Result Website

Result Search
- Search By Roll Number
- Search By Student Name
- Search By Father Name

Developed & Hosted by
Future Connect

Roll No	100001
Name	WAQAS HUSSAIN
Father's Name	MUNAWWAR HUSSAIN
Institute :	Private Student
Result	Fail
Remarks	Appeared as compartment case

FAIL RESULT

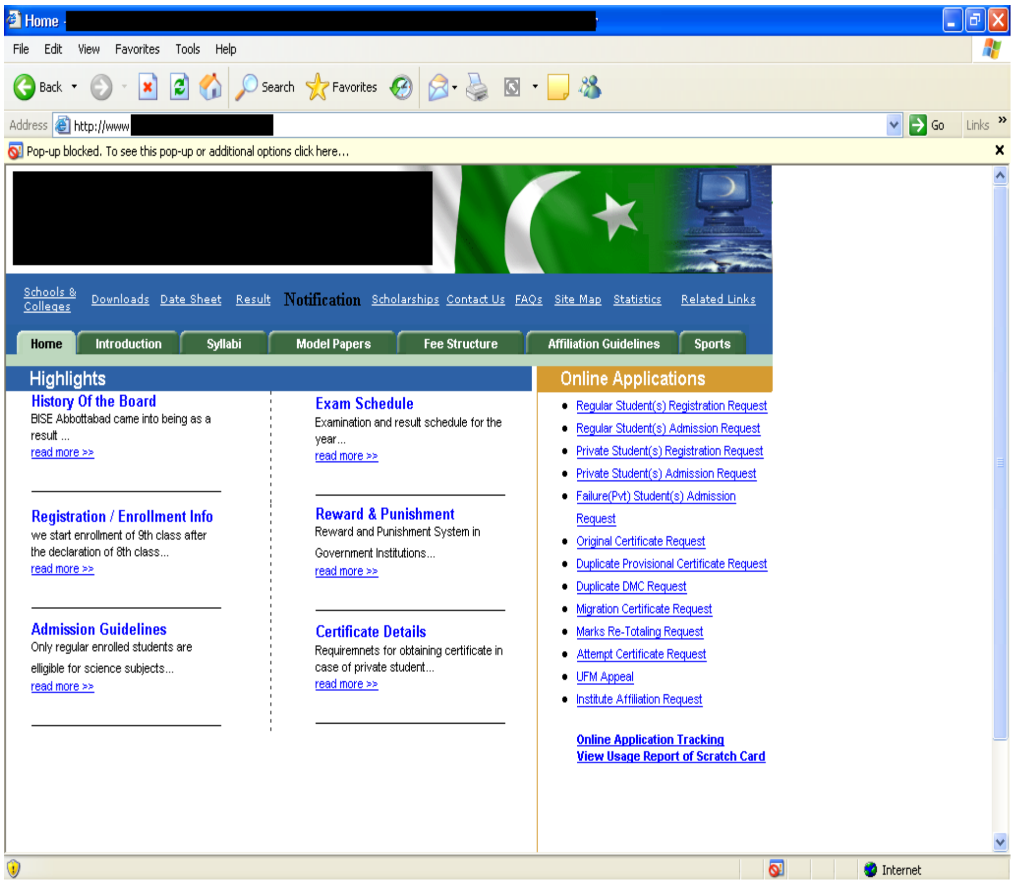
	SSC-I		SSC-II	
Subject	Theory	Practical	Theory	Practical
ENGLISH (COMPULSORY)	029	-	032	-
URDU (COMPULSORY)	038	-	046	-
PAKISTAN STUDIES	-	-	051	-
ISLAMIYAT	034	-	-	-
MATHEMATICS	025	-	008	-
PHYSICS	015	006	029	005
CHEMISTRY	022	008	020	009
BIOLOGY	021	009	025	013

Disclaimer :

This result gazette is issued provisionally, errors and omission excepted, as a notice only. Any entry appearing in this notification does not itself confer any right or privilege

Figure-2.6 Result of Waqas Hussain

3. Figure 9 shows the website which is the most vulnerable website I have come crossed.



Highlights

History Of the Board
BISE Abbottabad came into being as a result ...
read more >>

Registration / Enrollment Info
we start enrollment of 9th class after the declaration of 8th class...
read more >>

Admission Guidelines
Only regular enrolled students are elligible for science subjects...
read more >>

Exam Schedule
Examination and result schedule for the year...
read more >>

Reward & Punishment
Reward and Punishment System in Government Institutions...
read more >>

Certificate Details
Requiremnets for obtaining certificate in case of private student...
read more >>

Online Applications

- Regular Student(s) Registration Request
- Regular Student(s) Admission Request
- Private Student(s) Registration Request
- Private Student(s) Admission Request
- Failure(Pvt) Student(s) Admission Request
- Original Certificate Request
- Duplicate Provisional Certificate Request
- Duplicate DMC Request
- Migration Certificate Request
- Marks Re-Totaling Request
- Attempt Certificate Request
- UFM Appeal
- Institute Affiliation Request

Online Application Tracking
View Usage Report of Scratch Card

Figure-2.7 Another vulnerable website

2.8 FINDINGS

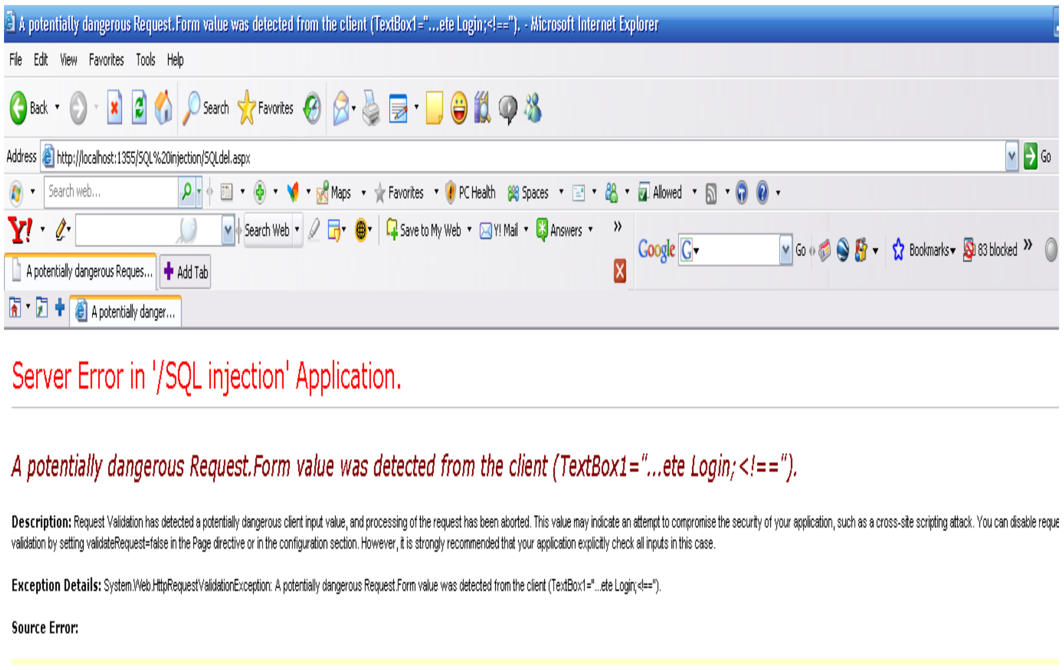
Software's are now becoming aware of it. .NET 2006 do identify SQL injections to some extend. If following query is at the back end.

string CommandText ="select * from faculty where first_name=‘ " + TextBox1.Text + “ ‘ ";

If following query is entered, that is an SQL injection attack.

‘;DELETE Login; <!==

Instead of going further it will be detected as an attack by .NET as shown in the figure.


A potentially dangerous Request.Form value was detected from the client (TextBox1="...ete Login;<!=="). - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:1355/SQL%20injection/SQLdel.aspx
Go

Server Error in '/SQL injection' Application.

A potentially dangerous Request.Form value was detected from the client (TextBox1="...ete Login;<!==").

Description: Request Validation has detected a potentially dangerous client input value, and processing of the request has been aborted. This value may indicate an attempt to compromise the security of your application, such as a cross-site scripting attack. You can disable reque validation by setting validateRequest=false in the Page directive or in the configuration section. However, it is strongly recommended that your application explicitly check all inputs in this case.

Exception Details: System.Web.HttpRequestValidationException: A potentially dangerous Request.Form value was detected from the client (TextBox1="...ete Login;<!==").

Source Error:

Figure-2.8 .NET error messages

2.7 TOOLS AND TECHNIQUES FOR PREVENTING SQL INJECTIONS

2.7.1 AUSELSQI Analysis and Design

AUSELSQL before forwarding the HTTP request to the web server it first intercepts the request as shown in the figure 3 .Following are the steps taken by AUSELSQL tool:

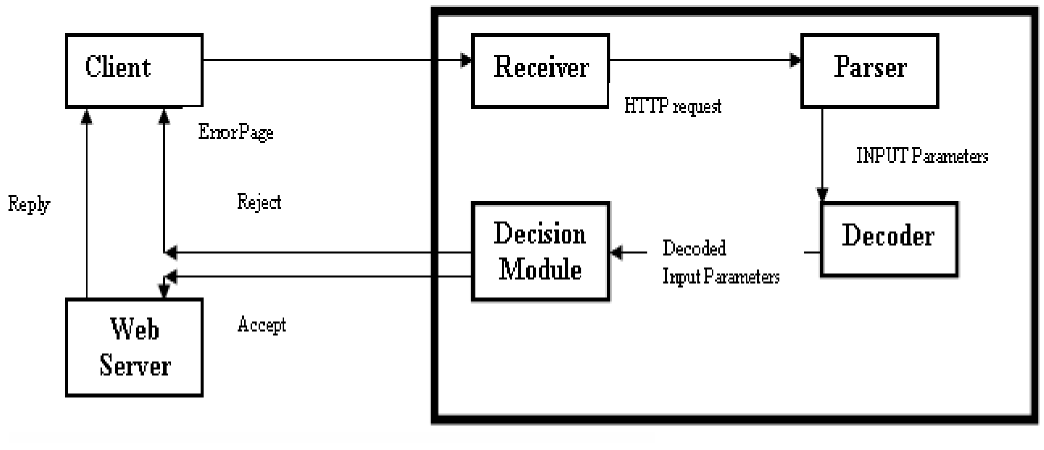


Figure-2.9 Data Flow diagram of the system [6]

1. **Receiver module**: It takes HTTP request from the client, intercepts it and then forwards it to the next step.
2. **Header processing module:** What it does, it takes input from the receiver module extract the users input and then pass it to decoder module.
3. **Decoder module**: Here encoded characters are transformed to ASCCII.

4. **Decision module**: Here decision making steps are taken place, this module look for the suspicious characters. If any illegal input is found it will not let it to proceed any further. But if it is legal, the client will be able to receive its request.

Limitations:-

1. It makes the system really slow.
2. Very complicated.
3. As it is implemented with the help of ISAPI filters that can not read the POST data. To read this data there is need to implement a method called ReadRawData. This method is called at very beginning phase of the receiving the request for the client. This will result the processing of the HTTP header in the filter code instead of depending on the already implemented code in IIS. This will result in double processing

2.7.2. AMNESIA Tool

The basic concept behind the AMNESIA tool is:

- static analysis
- Runtime monitoring.

It consists of four main steps.

1. Identify hotspots
2. Build SQL-query models
3. Instrument Application

4. Runtime monitoring

1 Identify hotspots: This step includes scanning of the whole application and identifying the hotspots. Hotspots are those areas where there are dynamic queries and they are vulnerable to SQL injections.

2 Build SQL-query models: Once all the hotspots are determined now in this step all possible quires will be generated as shown in the figure-4 and will be stored in the database**.**

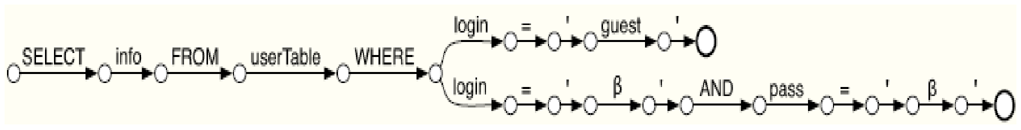


Figure-2.10 All possible out comes [9]

3 Instrument Application: Now at each hotspot call a function for runtime monitoring.

4 Runtime monitoring: In this step when ever dynamic query is called this Amnesia tool is called at run time then it check whether the query is legal or illegal by comparing it with the queries generated by SQL query model.

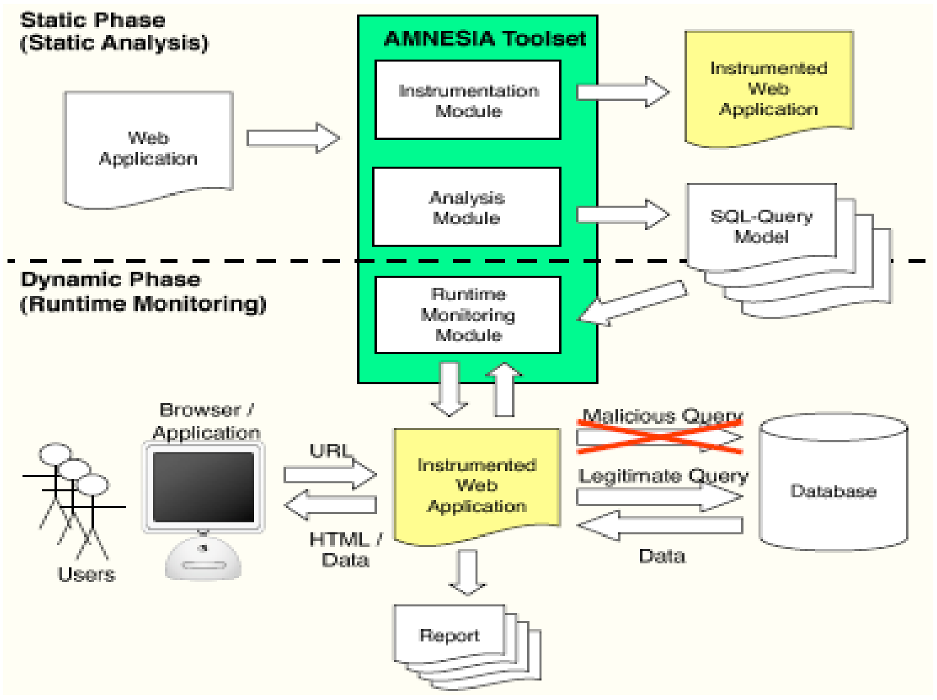


Figure-2.11 High level architecture of Amnesia tool [9]

Limitations:-

1. This is not realistic and difficult to achieve in big websites where a team of programmers collaborate in the website development.
2. Static analysis cannot determine the exact query at compile time.
3. It is a data intensive tool.

2.7.3. Parse Tree

In this method it will built two parse trees one with user input and the other without user input. Then compare the two parse tree for valid and invalid input. If it is valid then it will be precede further other wise it will not be proceed. Let's take an example We have a query Select * from users where username=? AND password=?

The figure-6 presents the parse tree without user input of the above query. Now when a user gives value to the SQL statement it will be validated by first making a parse tree as shown in the figure 7 & 8 then comparing it with the parse tree without user input. [10]

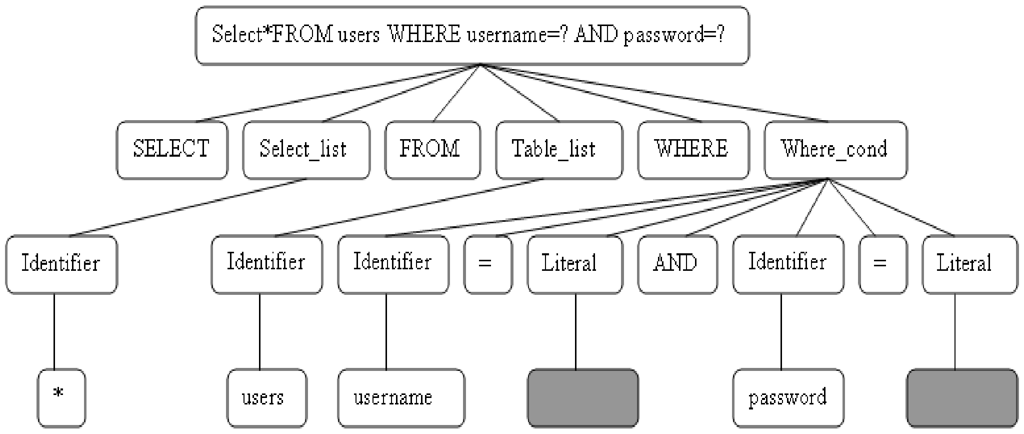


Figure-12 parse tree without user input

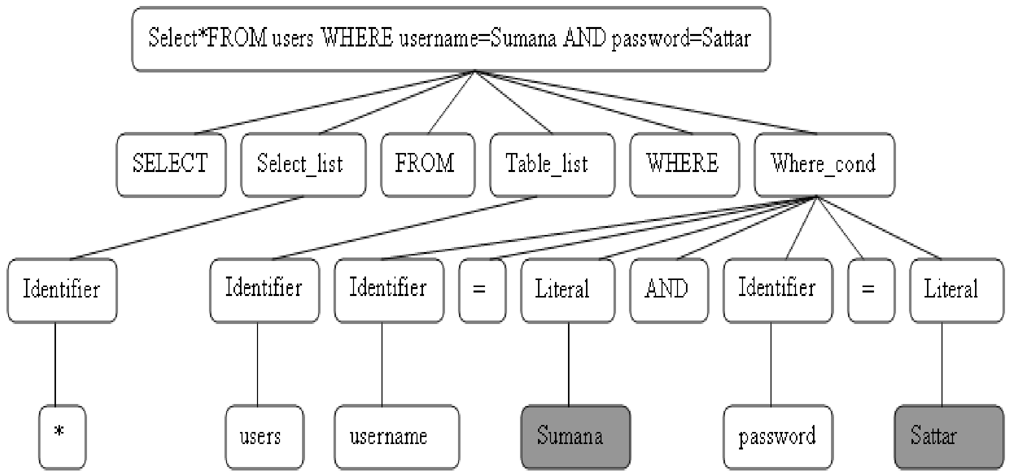


Figure-2.12 with valid user input

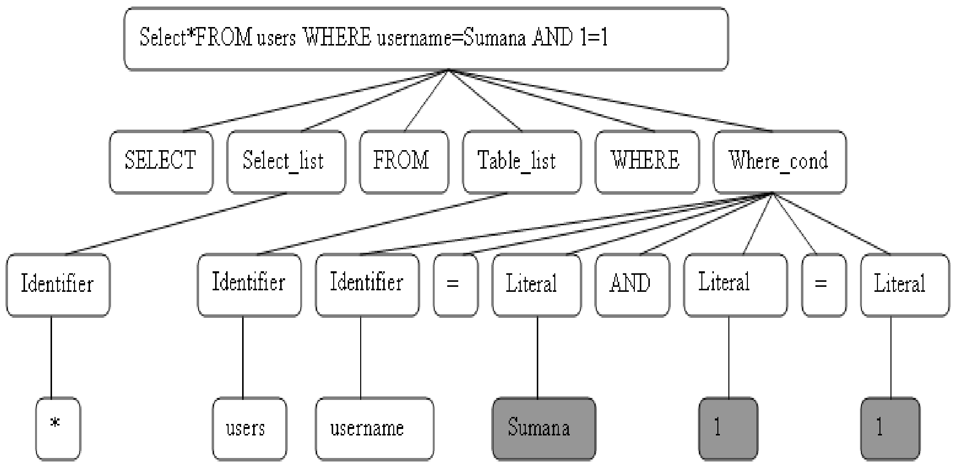


Figure-2.13 with invalid user input

Limitations

1. Time load is high but its performance is far better then amnesia or any other tool.
2. Doesn’t work efficiently when string is long and complicated.

Chapter 3

METHODOLOGY

To prevent SQL injections, parse tree technique was implemented. It was deployed on the website as web service, dll file and as a function call.

First identify all the dynamically generated queries on the website because these queries are potentially vulnerable to SQL injections. After that init() function is called on that query. Init function generates an encrypted key for the current thread. Then wrapper() function is called to wrap the input from the user. After wards classes safe_connection's and safe_statement's , extended by connection and statements, constructors are called. Where check() function of SQL_Guard are called. Here where parse tree technique is implemented, it will identify whether the query is valid or invalid, if it is valid then execute query function is called. Otherwise it will be consider as an SQL injection and it will not be allow going further.

Architecture diagram

First a user will enter values in the field; the whole dynamic query will be passed to the web server for the validity. Web server where there is implementation of the parse tree. Where two parse trees are made one with user input and other without user input. Both the trees will be compared if equal it will be consider as valid input other wise it will be consider as SQL injections and will not further be proceeded Figure-9 shows the architecture diagram of my project.

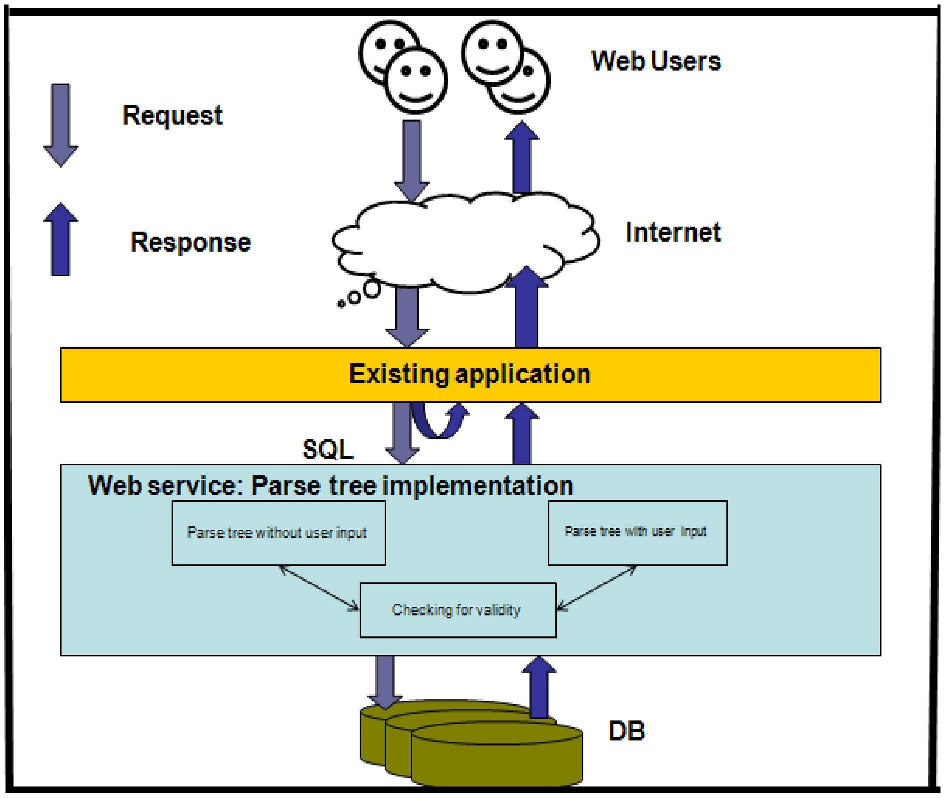

Request
Response
Web Users
Internet
Existing application
SQL
Web service: Parse tree implementation
Parse tree without user input
Parse tree with user input
Checking for validity
DB

Figure-3.1 Architecture diagram

3.2 CLASS DIAGRAM

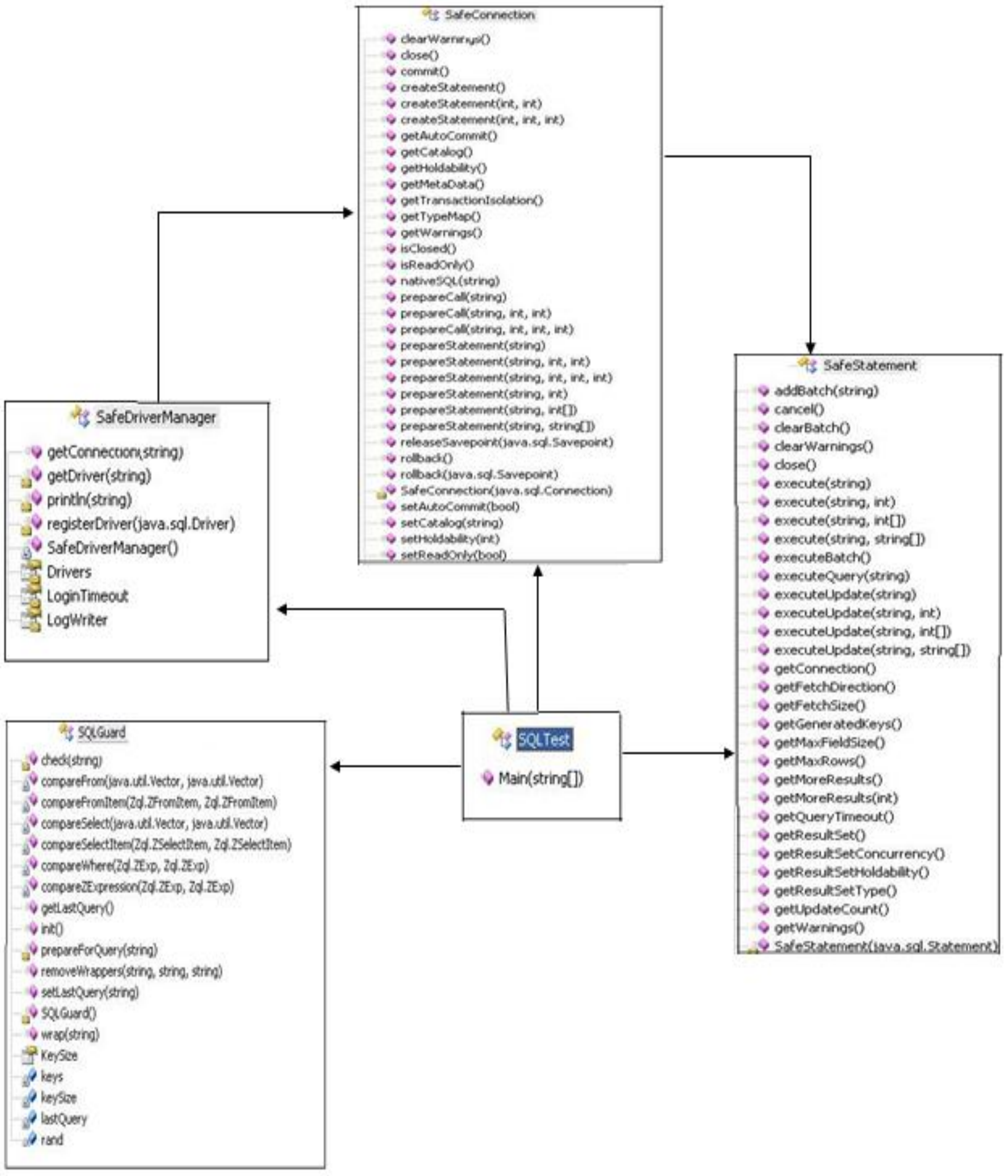


Figure-3.2 Class diagram

Use Case diagram

Use case is a set of scenarios that describe an interaction between a user and a system. A use case displays an interaction between actors and use cases.
An actor represents a user that will interact with the system. The use case diagram of my project is shown below.

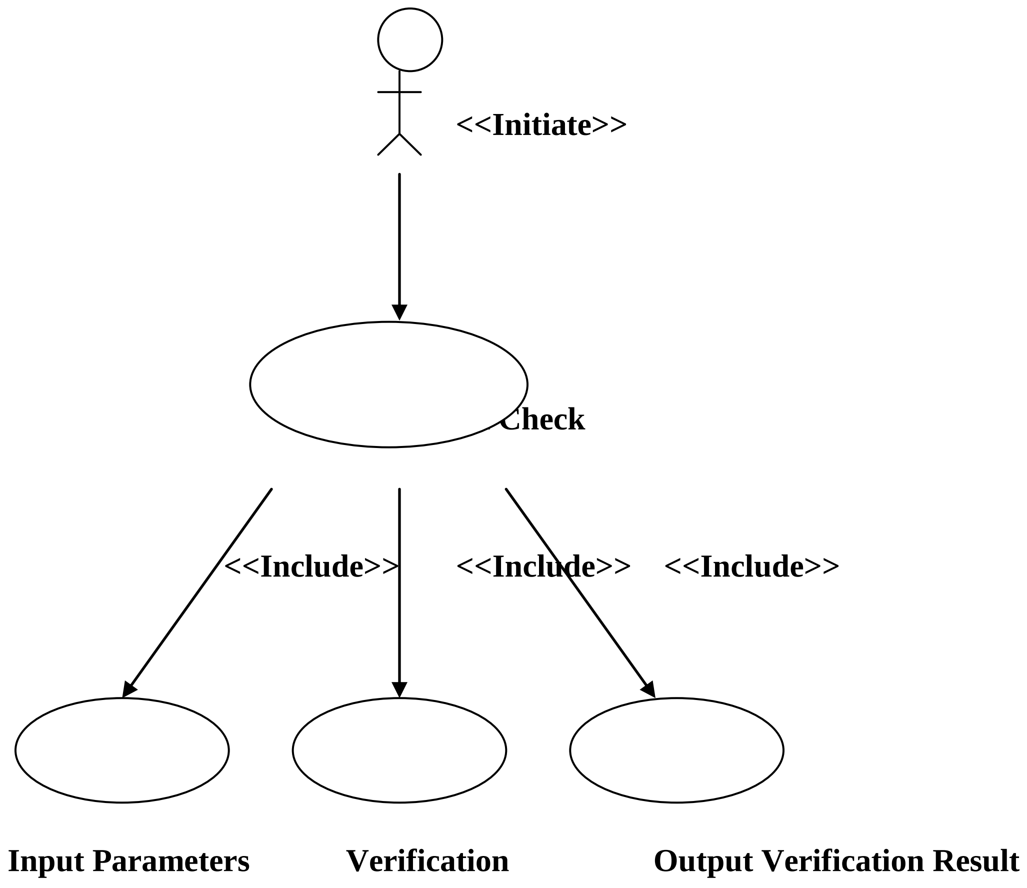


Figure-3.3 Use case diagram

Use case Name	SQL Injection check
Participating actors	User (hacker or any other user)
Flow of events	1. Input parameters 2. Verify whether the query is legal or an SQL injection attack. 3. If query is legal it will be allowed to go further if not it will not be allowed to gain access to the data base. 4. Out put verification result
Entry Condition	Input parameters
Exit Condition	Output verification result
Includes	Input Parameter, verification, output verification result
Extends	________

Table 1 Query use case flow of events

Chapter 4

IMPLEMENTATION

4.1 DFD

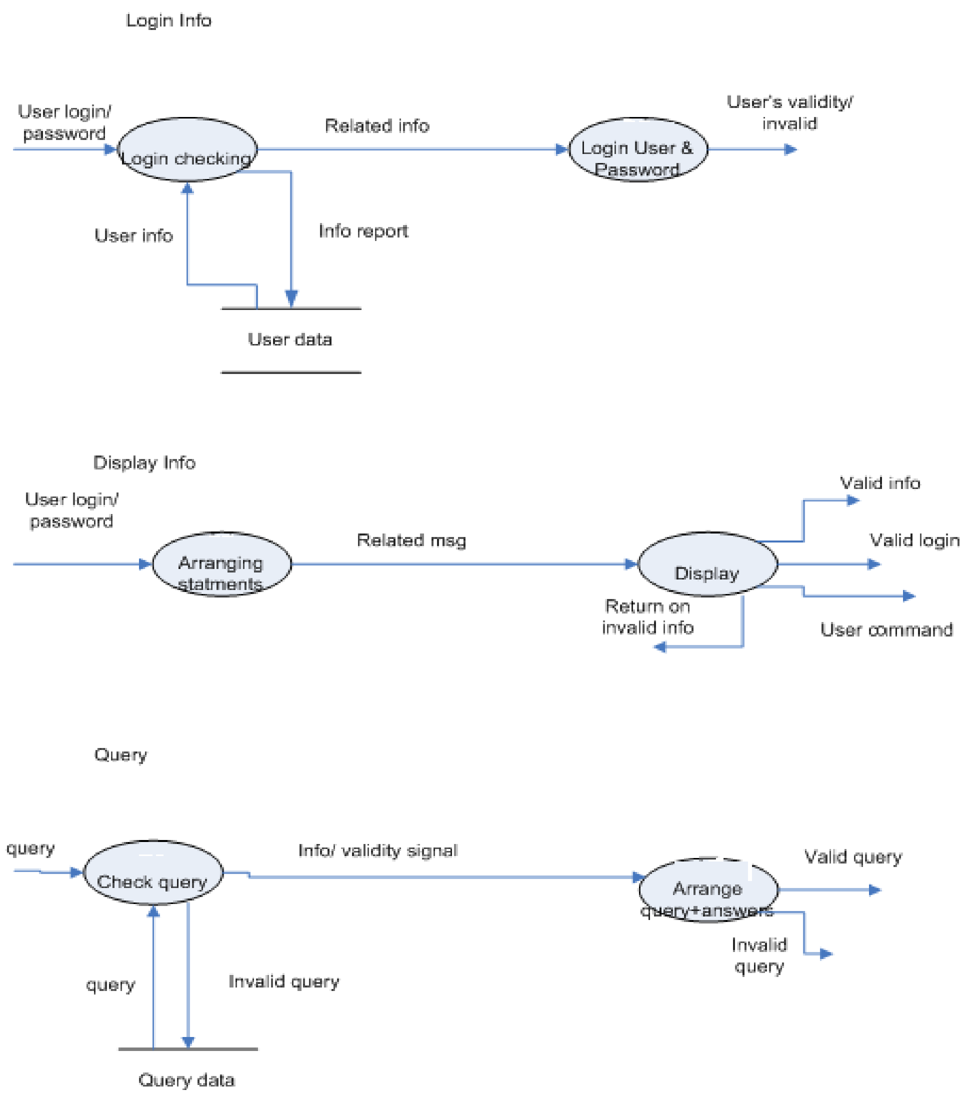


Figure-4.1 Data Flow Diagram

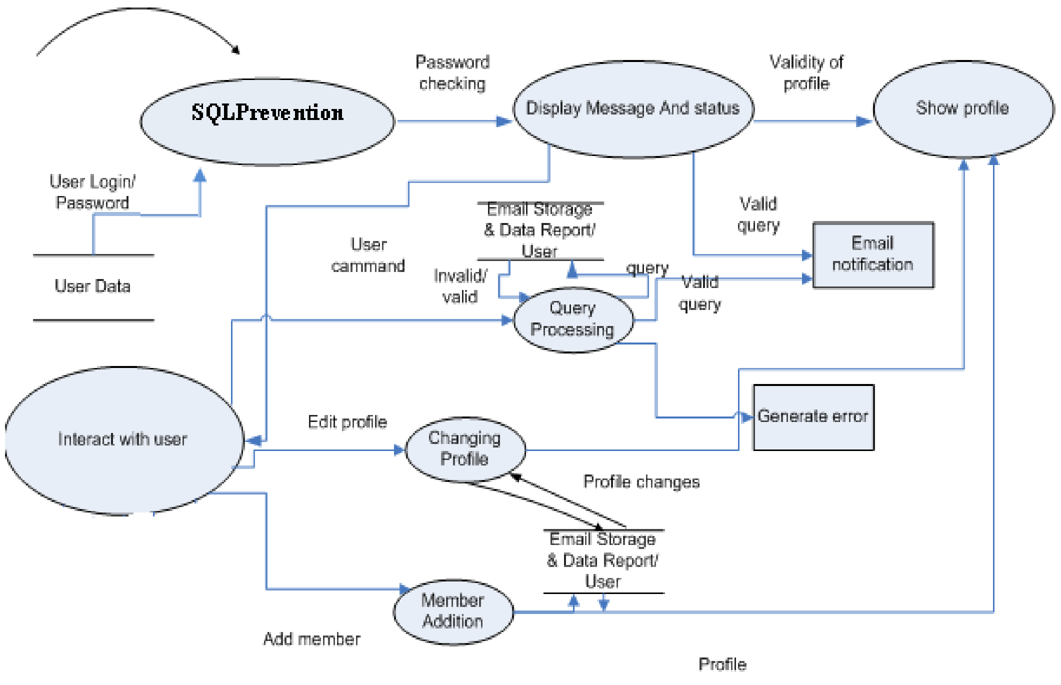


Figure-4.2 Data Flow Diagram

Explanation

The working flow of the system is that when a user enters the username and password to gain access to a system, it is checked by the SQL Prevention tool for SQL injection attacks. If the user enters the correct username and password, the user is then validated and is allowed to interact with the system e.g. to view data, send & receive email etc. If the user performs a SQL injection attack or enters an incorrect username or password, the query processing fails and an error is generated which is seen by the user as an invalid username or password message in the response notification.

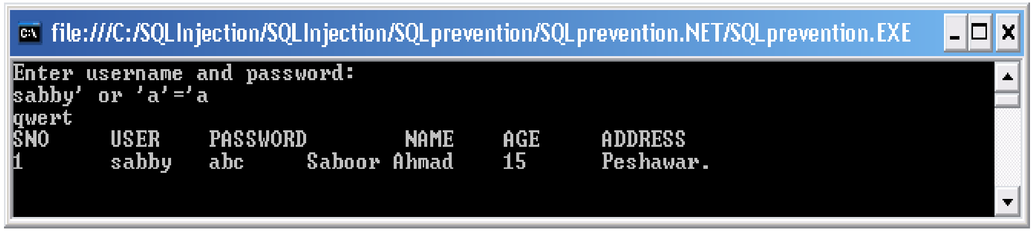
Chapter 5

TESTING

This chapter tests the software tool that is developed to prevent SQL injection. Firstly, the tool is tested on a console application to check its validity. Then it is integrated into a web application to be tested for the prevention of SQL injection.

5.1 TESTING SQL PREVENTION FOR CONSOLE APPLICATION

The tool to prevent SQL injection attacks was first tested on a simple console application. In the figure below it is shown that without the tool being utilized in the application, if a user appends a tautology after entering the username in the username field, the password becomes irrelevant and the user has an access to the record of the database. Even if the user enters an incorrect password, he can gain access to the record in the database.

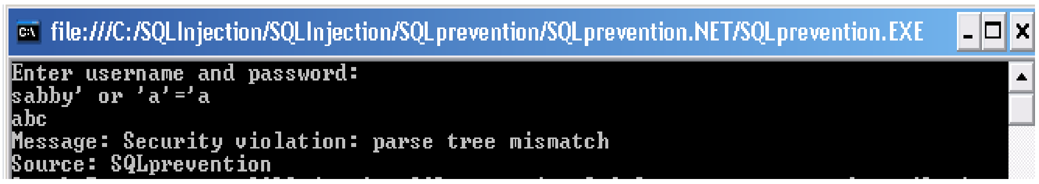

file:///C:/SQLInjection/SQLInjection/SQLprevention/SQLprevention.NET/SQLprevention.EXE

```
Enter username and password:
sabby' or 'a'='a
qwert
SNO     USER    PASSWORD        NAME    AGE     ADDRESS
1       sabby   abc     Saboor Ahmad    15      Peshawar.
```

Figure 5.1 SQL Injection attack

As seen from the above figure, the database is susceptible to SQL injection attacks. The user only enters the username along with an SQL injection tautology and gains access to the database. The actual password is ‘abc’ which becomes irrelevant due to SQL injection.

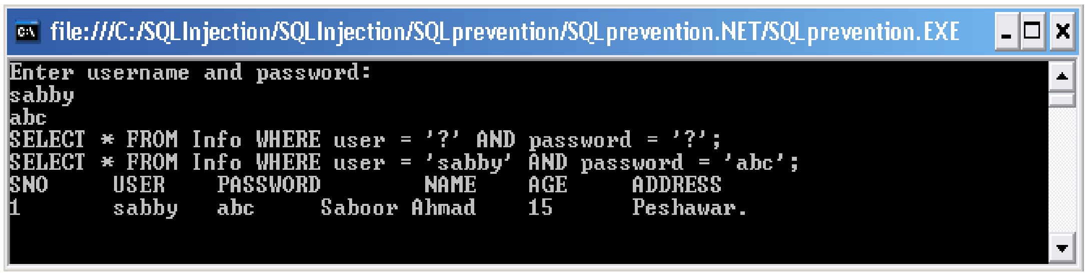
In order to prevent such attacks, the SQL prevention tool was developed. The following figure shows that an exception is generated if the parse tree of the intended input and the user input do no match.



```
file:///C:/SQLInjection/SQLInjection/SQLprevention/SQLprevention.NET/SQLprevention.EXE
Enter username and password:
sabby' or 'a'='a
abc
Message: Security violation: parse tree mismatch
Source: SQLprevention
```

Figure-5.2 Preventing SQL injection

So after integrating the SQL prevention tool with the application, the user can only gain access to the record if he enters the username and password correctly as can been from the figure below.

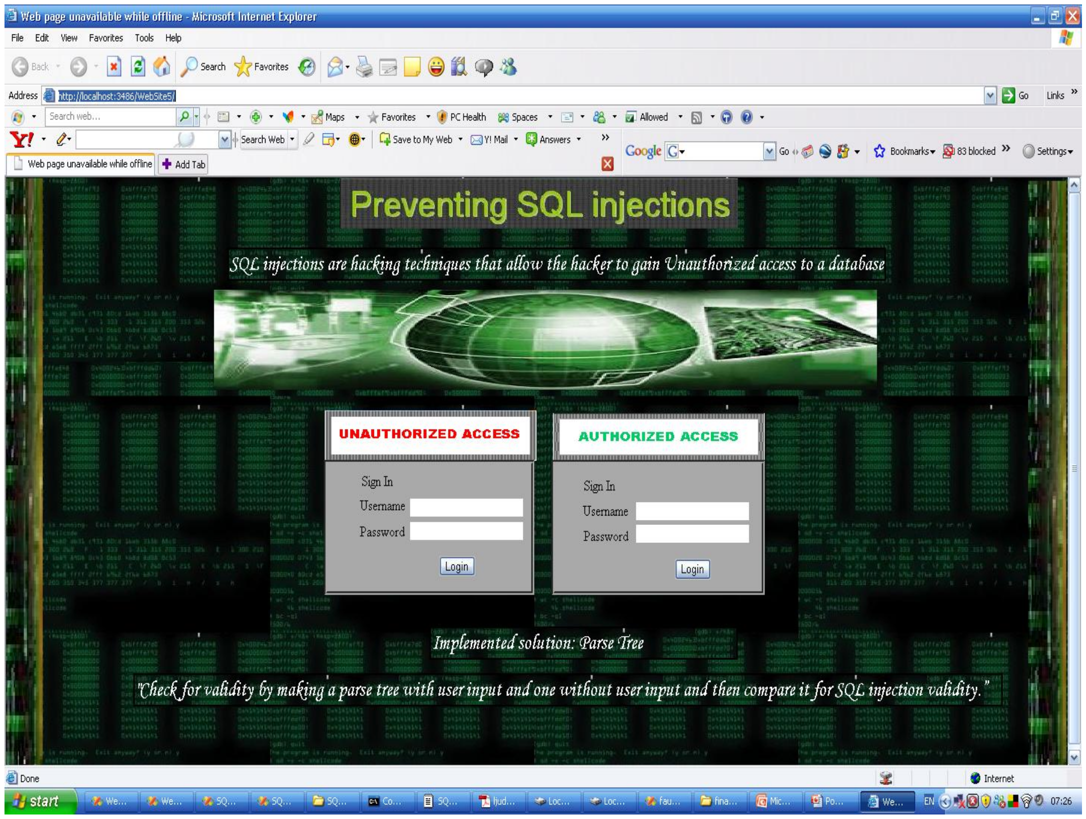


```
file:///C:/SQLInjection/SQLInjection/SQLprevention/SQLprevention.NET/SQLprevention.EXE
Enter username and password:
sabby
abc
SELECT * FROM Info WHERE user = '?' AND password = '?';
SELECT * FROM Info WHERE user = 'sabby' AND password = 'abc';
SNO     USER    PASSWORD        NAME    AGE     ADDRESS
1       sabby   abc     Saboor Ahmad    15      Peshawar.
```

Figure-5.3 Query processed after validation from SQL prevention tool

5.2 TESTING SQL PREVENTION FOR WEB APPLICATION

After the successful testing of the tool on the console application, it was then integrated into a web application. A simple web application of logging in a user with the correct username and password was developed. Figure 5.4 shows the interface of the web application with two parts left part is where tool is not deployed and right contain web service for the validity.


Web page unavailable while offline - Microsoft Internet Explorer
http://localhost:3486/WebSite5/
Preventing SQL injections
UNAUTHORIZED ACCESS
AUTHORIZED ACCESS
Sign In
Username
Password
Login
Done
Internet

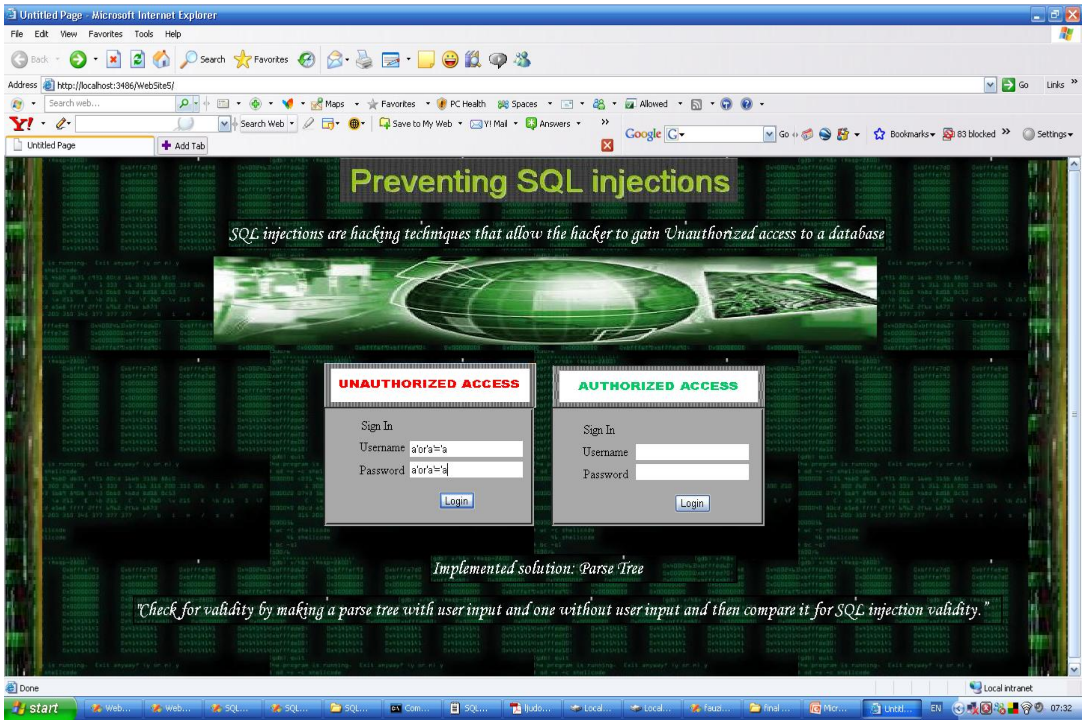
SQL injections are hacking techniques that allow the hacker to gain Unauthorized access to a database

Implemented solution: Parse Tree

"Check for validity by making a parse tree with user input and one without user input and then compare it for SQL injection validity."

Figure-5.4 Interface of web application

It can be seen from the figure below that without the SQL prevention tool being integrated into the application, the user can gain access to the website through an SQL injection attack.



Untitled Page - Microsoft Internet Explorer

Address http://localhost:3486/WebSite5/

Preventing SQL injections

SQL injections are hacking techniques that allow the hacker to gain Unauthorized access to a database

UNAUTHORIZED ACCESS

Sign In

Username a'or'a'='a

Password a'or'a'='a

AUTHORIZED ACCESS

Sign In

Username

Password

Implemented solution: Parse Tree

"Check for validity by making a parse tree with user input and one without user input and then compare it for SQL injection validity."

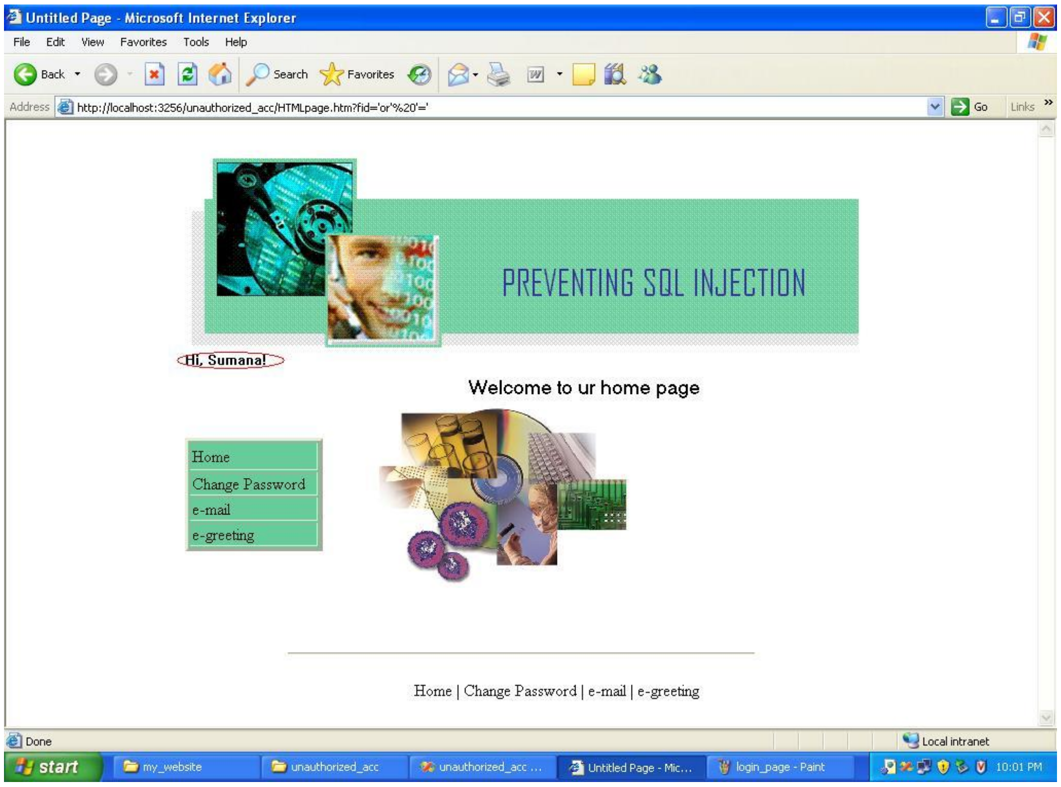
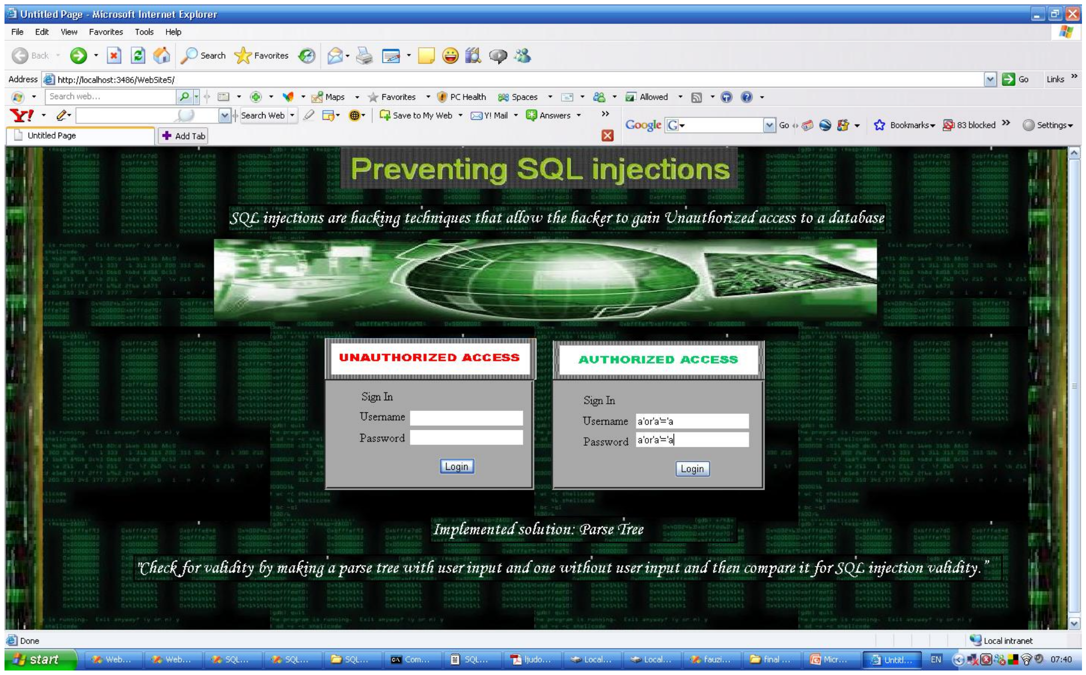

Untitled Page - Microsoft Internet Explorer
http://localhost:3256/unauthorized_acc/HTMLpage.htm?fid='or'%20'='
PREVENTING SQL INJECTION
Hi, Sumana!
Welcome to ur home page
Home
Change Password
e-mail
e-greeting
Home | Change Password | e-mail | e-greeting
Done
Local intranet

Figure-5.5 SQL injection attack on web application

After implementing the SQL prevention tool into the web application, the user cannot enter the website through SQL injection attack as shown below.


Untitled Page - Microsoft Internet Explorer
http://localhost:3486/WebSite5/
Preventing SQL injections
SQL injections are hacking techniques that allow the hacker to gain Unauthorized access to a database
UNAUTHORIZED ACCESS
AUTHORIZED ACCESS
Sign In
Username
Password
Login
Sign In
Username a'or'a'='a
Password a'or'a'='a
Login
Implemented solution: Parse Tree
"Check for validity by making a parse tree with user input and one without user input and then compare it for SQL injection validity."

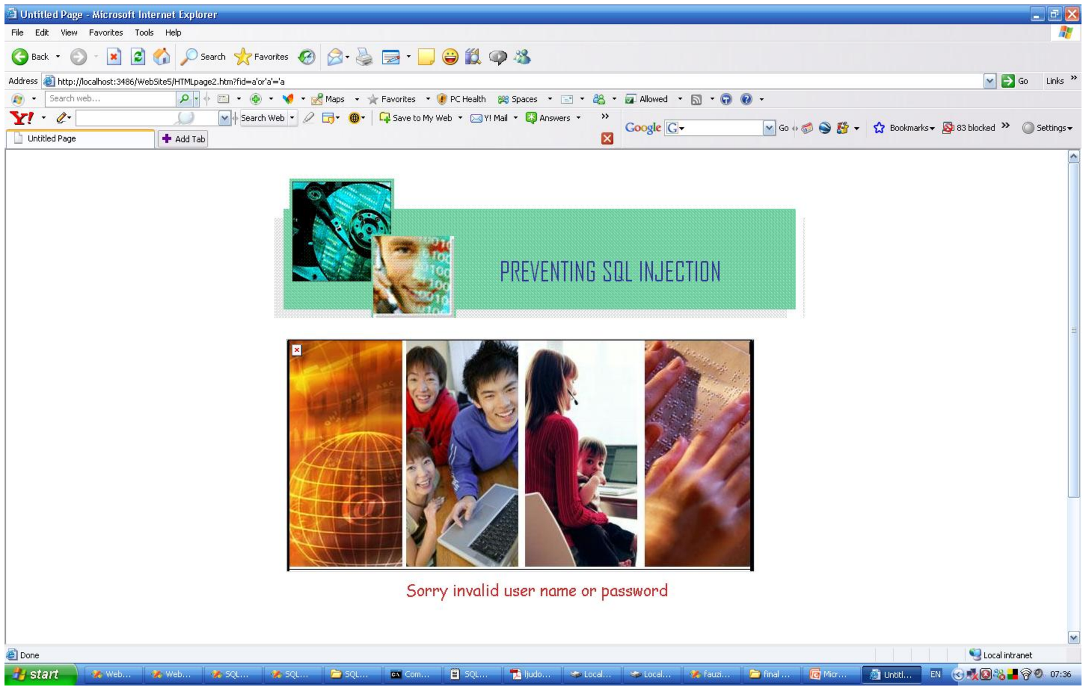
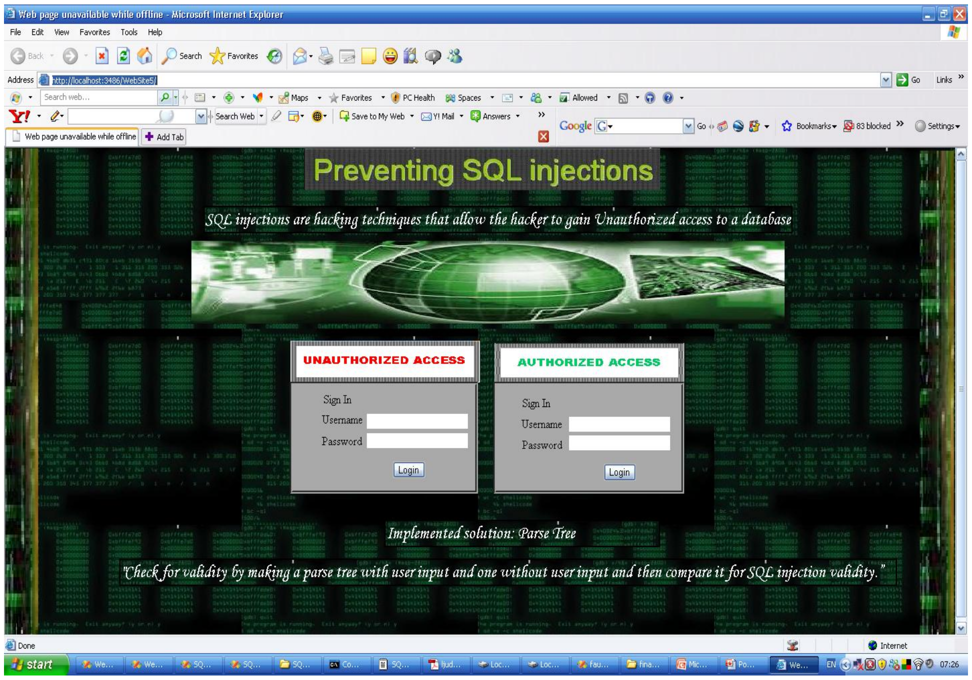

Untitled Page - Microsoft Internet Explorer
http://localhost:3486/WebSite5/HTMLpage2.htm?id=a'or'a'='a
PREVENTING SQL INJECTION
Sorry invalid user name or password

Figure-5.6 – Preventing SQL injection on web application

A user can only gain access to the website if he enters a valid username and password that is parsed by the SQL prevention tool and the SQL query is then executed.


Web page unavailable while offline - Microsoft Internet Explorer

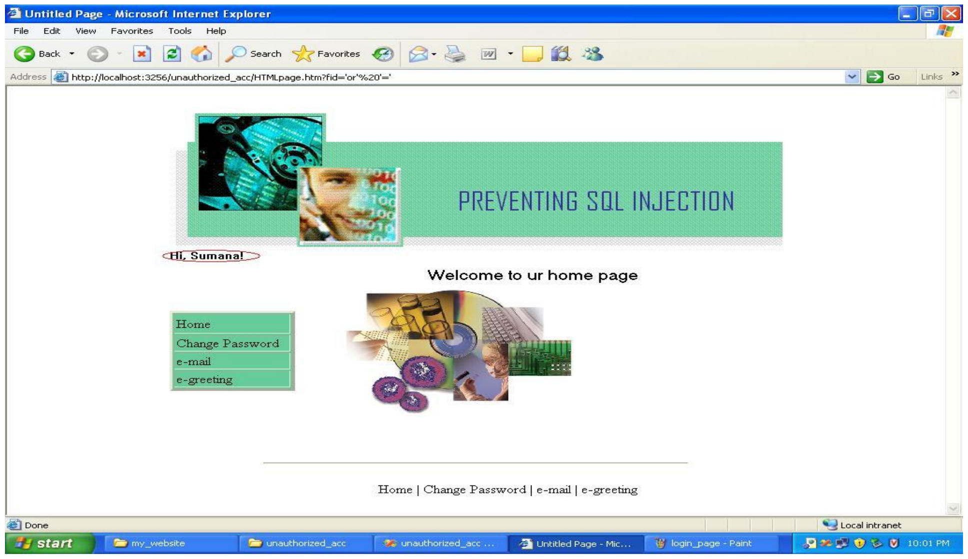
Address http://localhost:3486/WebSite5/

Preventing SQL injections

SQL injections are hacking techniques that allow the hacker to gain Unauthorized access to a database

Implemented solution: Parse Tree

"Check for validity by making a parse tree with user input and one without user input and then compare it for SQL injection validity."


Untitled Page - Microsoft Internet Explorer

Address http://localhost:3256/unauthorized_acc/HTMLpage.htm?fid='or'%20'='

PREVENTING SQL INJECTION

Hi, Sumana!

Welcome to ur home page

Figure-5.7 Access granted after validation from SQL prevention tool

Chapter 6

CONCLUSION

SQL injections are hacking techniques which are used by hackers to gain unauthorized access to database. The hacker takes advantage of common programming blunders made by programmers. Actually when invalid input is entered by hacker it is not validated first and is allowed to proceed further. Lot of information is available over the net and any one can learn and master in this. It was because of this reason that in 2006 it was declare as 2nd biggest threat to web base application. During this project I learn about SQL injections and checked different websites for the vulnerabilities. And many websites were found vulnerable to SQL injections. During this I came across a very interesting thing that now software's like .NET 2006 are becoming aware of it. But they are not completely stopping the SQL injections. Many solutions have been proposed to prevent SQL injections but with some limitation. Parse, which is implemented in this project is the best solution for the prevention of SQL injections. It is following a very simple rule. It makes a parse tree without user input and parse tree with user input, compare both to check for validity. If equal it will be considered as safe query and will be allowed to enter the database otherwise it will not be allowed to gain access to the data base. Parse tree has some limitations. It does not work efficiently for complicated SQL queries. I deployed this tool on my website and then tried the SQL injections; it completely prevents the SQL injections.

REFRENCES

1. "Blind SQL Injection" , Dec,12 2006, <http://www.securitydocs.com/library/2651>
2. "SQL Injection, are your web application vulnerable?", Dec,13 2006, <http://www.securitydocs.com/library/2656>
3. "SQLBlock: SQL Injection Protection by Variable Normalization of SQL Statement", Dec 14 2006, <http://www.securitydocs.com/library/3388>
4. "SQL Injection", Feb 15 2007, <www.imperva.com/application_defense_center/glossary/sql_injection.html>
5. "SQLInjectionWhitePaper", Feb 16 2007, <www.spidynamics.com/papers/SQLInjectionWhitePaper>
6. Maor Ofer, Shulman Amichai, "Blind Folded SQL injections", RSA Conference 2005.
7. "Blind_SQLInjection", March 15 2007 , <www.spidynamics.com/whitepapers/Blind_SQLInjection>
8. "The myth of SQL Injections", March 10 2007 , <http://searcsecurity.techtarget.com/tip/0,289483,sid14_gci1193902,00.html>
9. William G.J. Halfond, Alessandro Orso, "Preventing SQL Injections using AMNESIA" ,International Conference on Software Engineering – ICSE
10. Gregory T. Buehrer, Bruce W. Weide, Paolo A. G. Sivilott, "Using Parse Tree Validation to Prevent SQL Injection Attacks", ACM Conference, Columbus 2005.