ALL YOUR USERS ARE OUT TO GET YOU
(Okay, not all, but enough)
SQL Injection Attacks Overview

Malicious users send input to forms that attempts to get more information than you intended or alter the database in some way.

Possible impacts: Loss of sensitive data (UCLA), denial of service attacks, web page vandalism

An automated attack only needs to have a 1/10,000 chance in succeeding to be feasible.
Sounds difficult. . .right?

Not validating user input is one of most common mistakes in programming in general – SQL injection attacks take advantage of this flaw. They're easy to do and easy to automate.
Prove it!

Web Goat is a fun demonstration of various web page security concerns. We're going to concentrate on SQL injection.

Web Goat is a Tomcat web server that's vulnerable to SQL injection attacks (among others). It's available at: http://www.owasp.org/index.php/Category:OWASP_P_WebGoat_Project
Web Goat
Lesson: How to Perform Blind SQL Injection

Let's go through this lesson together.
Blind SQL Injection (1)

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the first_name in table user_data for userid 15613. Put that name in the form to pass the lesson.

Enter your Account Number: 101

Account number is valid

By Chuck Willis

OWASP Foundation | Project WebGoat
Blind SQL Injection (2)

![Tomcat output showing command execution and web requests]

```
\WebGoat-5.0\tomcat\webapps\WebGoat\lesson_plans
Executing OS command: cmd.exe /c type "C:\Documents and Settings\Megan\Desktop\WebGoat-5.0\tomcat\webapps\WebGoat\lesson_plans\BasicAuthentication.html"
 - WebGoat: Wed May 09 11:35:45 CDT 2007; 127.0.0.1:127.0.0.1; org.owasp.webgoat.lessons.CommandInjection; [Screen=2,menu=610]
Wed May 09 11:35:45 CDT 2007; 127.0.0.1:127.0.0.1; org.owasp.webgoat.lessons.CommandInjection; [Screen=2,menu=610]
Executing OS command: cmd.exe /c dir /b "C:\Documents and Settings\Megan\Desktop\WebGoat-5.0\tomcat\webapps\WebGoat\lesson_plans"
Executing OS command: cmd.exe /c type "C:\Documents and Settings\Megan\Desktop\WebGoat-5.0\tomcat\webapps\WebGoat\lesson_plans\AccessControlMatrix.html"
 - WebGoat: Wed May 09 11:36:02 CDT 2007; 127.0.0.1:127.0.0.1; org.owasp.webgoat.lessons.CommandInjection; [HelpFile=AccessControlMatrix.help,SUBMIT=View,menu=610]
Wed May 09 11:36:02 CDT 2007; 127.0.0.1:127.0.0.1; org.owasp.webgoat.lessons.CommandInjection; [HelpFile=AccessControlMatrix.help,SUBMIT=View,menu=610]
 - WebGoat: Wed May 09 11:37:01 CDT 2007; 127.0.0.1:127.0.0.1; org.owasp.webgoat.lessons.BlindSqlInjection; [Screen=46,menu=610]
Wed May 09 11:37:01 CDT 2007; 127.0.0.1:127.0.0.1; org.owasp.webgoat.lessons.BlindSqlInjection; [Screen=46,menu=610]
 - WebGoat: Wed May 09 11:37:45 CDT 2007; 127.0.0.1:127.0.0.1; org.owasp.webgoat.lessons.BlindSqlInjection; [Submit=Go!,menu=610,account_number=101]
Wed May 09 11:37:45 CDT 2007; 127.0.0.1:127.0.0.1; org.owasp.webgoat.lessons.BlindSqlInjection; [Submit=Go!,menu=610,account_number=101]
```
Blind SQL Injection (3)

Is this app vulnerable? Yes! What's wrong with this response?

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true/false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the first_name in table user_data for userid 15613. Put that name in the form to pass the lesson.

Enter your Account Number: 'or1=1;-- Go!

An error occurred, please try again.

By Chuck Willis
Blind SQL Injection (4)

What will the admin see?
Blind SQL Injection (5)

What if I want to get the userid associated with this account number?
Blind SQL Injection (6)

It would take a long time to try each number. *But we don't need to.*

Compound SQL statements can be made by joining multiple tests with keywords like AND and OR. Create a SQL statement that you can use as a true/false test and then select the first character of the target element and do a start narrowing down the character using > and <

The backend database is Microsoft Access. Keep that in mind if you research SQL functions on the Internet since different databases use some different functions and syntax.

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the first_name in table user_data for userid 15613. Put that name in the form to pass the lesson.

Enter your Account Number: 101 and userid<125

Account number is valid

By Chuck Willis
Blind SQL Injection (7)

Compound SQL statements can be made by joining multiple tests with keywords like AND and OR. Create a SQL statement that you can use as a true/false test and then select the first character of the target element and do a start narrowing down the character using > and <

The backend database is Microsoft Access. Keep that in mind if you research SQL functions on the Internet since different databases use some different functions and syntax.

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the first_name in table user_data for userid 15613. Put that name in the form to pass the lesson.

Enter your Account Number: 101 and userid<80  Go!

Invalid account number

By Chuck Willis
Blind SQL Injection (8)

A couple guesses later. . .

Compound SQL statements can be made by joining multiple tests with keywords like AND and OR. Create a SQL statement that you can use as a true/false test and then select the first character of the target element and do a start narrowing down the character using > and <

The backend database is Microsoft Access. Keep that in mind if you research SQL functions on the Internet since different databases use some different functions and syntax.

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the first_name in table user_data for userid 15613. Put that name in the form to pass the lesson.

Enter your Account Number: 101 and userid=101

Account number is valid

By Chuck Willis
Blind SQL Injection (9)
But what about a different user? And can I find out more?

Compound SQL statements can be made by joining multiple tests with keywords like AND and OR. Create a SQL statement that you can use as a true/false test and then select the first character of the target element and do a start narrowing down the character using '>' and '<'.

The backend database is Microsoft Access. Keep that in mind if you research SQL functions on the Internet since different databases use some different functions and syntax.

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true/false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the first_name in table name in the form to pass the lesson.

Enter your Account Number: 
Account number is valid
Blind SQL Injection (10)

Guessing some more... 

'Joesph' has a userid of 15613 and an account number of 15613. 10 minutes of work for a human. Much less for an automated attacker.
But that's just information leakage.

Ask UCLA about how bad that can be. But even beyond that there's more we can do with just this attack.

The magic words?

'a' or '1'='1

(Magic words vary by application and database.)
LAB: SQL Injection

Let's go through this lesson together.
Stage 1: Use String SQL Injection to bypass authentication. The goal here is to login as the user Neville Bartholomew, who is in the Admin group. You do not have the password, but the form is SQL injectable.

* Login failed
Stage 1: Use String SQL Injection to bypass authentication. User Neville Bartholomew, who is in the Admin group, is shown the login form is SQL injectable.

* Login failed
LAB: SQL Injection (3)

Foiled by the limit set on the number of letters in the field. That's ok. We can get around that.
LAB: SQL Injection (4)

Larry isn't all that important. So what?
This is just two examples of SQL injection attacks. These attacks can accomplish anything SQL can do. Do you have a form that just spits back the results of a certain query? How about a form that accepts credit cards? What about a form that charges for things based on the price in the database?

Are your logs vulnerable to SQL injection attacks?
Defenses

(A very abridged list.)
Check your logs regularly to see if any attempts have succeeded. Make sure they're able to record such attacks.

Practice good programming – validate user input and always filter out characters that aren't needed. (Will a name ever include a '%'?)

Limit the rights of the user that runs the queries for the web form to the minimum necessary.
Rest of time

Go through more of the lessons in Web Goat.