Colin Passno, LaVonne Patoir

CEN 4025c

Professor Walauskis

4/8//2025

**XSS Injection Exploit in Webapp**

**Concept:** Cross-site scripting, or XSS, is a form of injection exploit that allows a threat actor to force a webpage’s visitors to execute malicious code. In our particular example, a user inputs a string into a field that web browsers will interpret as HTML code when the page loads.

**Vulnerability:** Our program was originally vulnerable to this exploit because a field that accepted user input was saved to the database without any modification. When the program then generated an HTML page for a browser to view, this string was interpreted as HTML code that executed a script when the page was loaded. In our example, this script redirects the browser to an external site.

**Solution:** A small function is added to the ToDoList object class that replaces special characters with their HTML encoding equivalent, such as "&" becoming "&amp;". This one small step prevents a browser from parsing a user-provided string as code, and as such prevents the exploit. Please view the included video file to see the effects of a malicious string pre and post-fix.

**Old ToDoList.java:**

package com.example;

import jakarta.persistence.\*;

@Entity

@Table(name = "ToDoListItems")

public class ToDoList {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column(nullable = false)

private String task;

public ToDoList() {}

public ToDoList(String task) {

this.task = task;

}

public int getId() {

return id;

}

public String getTask() {

return task;

}

}

**New ToDoList.java:**

package com.example;

import jakarta.persistence.\*;

@Entity

@Table(name = "ToDoListItems")

public class ToDoList {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column(nullable = false)

private String task;

public ToDoList() {}

public ToDoList(String task) {

this.task = scrub(task);

}

public int getId() {

return id;

}

public String getTask() {

return task;

}

private String scrub(String input) {

if (input == null) {return null;}

return input.replace("&", "&amp;")

.replace("<", "&lt;")

.replace(">", "&gt;")

.replace("\"", "&quot;")

.replace("'", "&#x27;");

}

}

**Malicious Code:**

"/><script type="text/javascript">window.onload = function() {window.location.href = "https://www.youtube.com/watch?v=xvFZjo5PgG0";};</script>