Evgenii Litvinov

Professor Aron Laszka

COSC 3371 / 2021 Spring

April 23, 2021

Homework 4

Problem 1 (2 points): SQL Injection

1. ‘;INSERT INTO users (username,password) VALUES (‘evgenii’,’evgenii’);--

2. The first step to preventing this attack is validating user inputs. For example, input fields for email addresses can be filtered to allow only the characters in an email address, such as a required “@” character. Secondly, Sanitize Data by limiting special characters. For example, you can't use unfamiliar characters as "(",")"

Problem 2 (2 points): Cross-Site Scripting

1. <s<scriptcript>alert(document.cookie)</s<scriptcript>

2. The first step to preventing this attack is to Validate and filter all input data on the webserver. Also, the filter must be applied repeatedly until nothing is left to transform. Second, remove or encode all special HTML characters.

Problem 3 (2 point): File Inclusion

1. http://192.168.1.83/header.php?theme=../../../etc/passwd

2. The first step to preventing this attack is to prevent users from passing input into the file systems and framework API. Second, always validate input before using it, do not use values directly.

Problem 4 (2 point): Command Injection

1. http://192.168.1.83/gallery.php?removeImage=;cat/etc/passwd

2. The first step to preventing this attack is do not allow any user input to command your application is executing.

Problem 5 (2 point): File Upload

1. http://192.168.1.83/gallery.php?theme=images/upload489127254.jpg

2. I use file upload vulnerability where an application allows a me to upload a malicious file directly which is then executed. We can use the uploaded script to display /etc/passwd by doing command injection from problem 4.