Lab 4: Input Validation

QUICK REVIEW

Hackers attack websites by sending malicious inputs. This could be through a web form, or by sending requests directly to your API with tools, or by using an intercepting proxy. Input validation means to check on the server side that the input supplied by the user/attacker is of the form that you expect it to be. If it is not of the right form, then the data should be rejected. In other words, Input validation, also known as data validation, is the proper testing of any input supplied by a user or application. Input validation prevents improperly formed data from entering an information system. Because it is difficult to detect a malicious user who is trying to attack software, applications should check and validate all input entered into a system. Input validation should occur when data is received from an external party, especially if the data is from untrusted sources. Incorrect input validation can lead to injection attacks, memory leakage, and compromised systems.

In this lab, we will look at some basic practices of input validation.

Important Notice:

Please carefully read the disclaimer declaration on the course webpage, before you start the lab practice, and make sure you fully understand all statements. The disclaimer is available on https://hogeschool.github.io/Software-Quality.

LAB PRACTICES

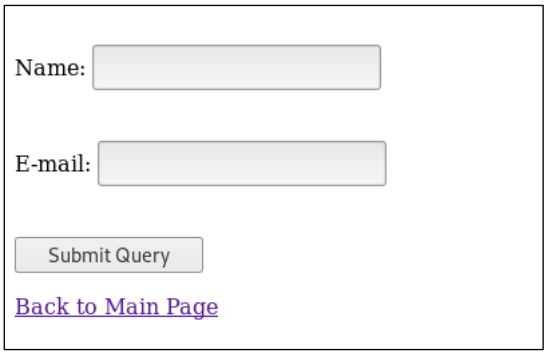
To setup the practices for this Lab, please download the zip file below from GitHub and extract it to your "dvwa" folder on Kali Linux: https://hogeschool.github.io/Software-Quality/labs/input-validation.rar

Start the web server in your Kali Linux: `service apache2 start`, and go to the link below:

```
http://127.0.0.1/dvwa/input-validation
```

4.1. A Simple Form (No Input Validation)

Click on the link provided for this practice to see the following simple form.



4.1.1. Check the Vulnerability

Check the form and examine whether any of these two fields are being validated for user input or not? Discuss your findings in the group.
Is the required field examined by the application?

4.1.2. Examine more Scripts

As you realized both Name and E-mail fields are not validated for input. Hence, they can be misused by attacker to enter his/her scripts. Enter the following scripts (one by one) and observe the consequences:

```
<h1>John</h1>
<font color=”green”>John</font>
<script> alert("Your System is Hacked!"); </script>
```

Try your own simple scripts and discuss the results in your group.

4.1.3. More Practices

Now, you can try more advanced scripts and analyse the results. There are two more suggested exercise below:

- Enter the following script and observe the result:

```
<script> document.location.replace("http://www.autogradr.com"); </script>
```

- Enter a script to open 5 tabs to show a desired page, such as www.google.com. You can check the textbook to figure out how to write the script.
- Enter the following script and observe the result:

```
<script> document.body.innerHTML = document.body.innerHTML.replace('Form Validation', 'Your
Page is Hacked!'); </script>
```

4.2. A Form with Low Level of Input Validation

Back to Main Page and click on the link provided for “Form Validation (Low)”.
Repeat all previous practices on this form and compare the results with the previous results.
Analyse and discuss your finding in the group.

4.3. A Form with Medium Level of Input Validation

Back to Main Page and click on the link provided for “Form Validation (Medium)”.
Repeat all previous practices on this form and compare the results with the previous results.
Analyse and discuss your finding in the group.

4.4. A Form with High Level of Input Validation

Back to Main Page and click on the link provided for “Form Validation (High)”.
Repeat all previous practices on this form and compare the results with the previous results.
Analyse and discuss your finding in the group.

4.5. Code Analysis

At this part, we will take a deeper look at the codes of these forms. You can find the codes in the following subfolder of your “`dvwa`” installation folder: `input-validation/`

4.5.1. Check the Input Validation Code

There are three level of input validation in three PHP files. Carefully investigate each and compare them to find the differences among them.
Discuss the findings in your group.

4.5.2. Still Attackable?

Open the “Form Validation (High Security Level)” page, and add the following text to the URL, and press Enter:

```
/%22%3E%3Cscript%3Ealert('hacked')%3C/script%3E
```

What is the result? Discuss it in your group.