Security Testing Project Final Report Carlo Fanciulli 198793

Introduction

This is the final report for the security testing examination of the 13th June 2020. This project consists in analyzing the php code of an ecommerce platform called Inventory-Management-System. Then, the Pixy tool was used to detect XSS vulnerabilities and classify them True Positive or False Positive. For True Positive cases are written automated test cases using the Selenium tool to assert the presence of the vulnerabilities. Finally, the vulnerabilities have been fixed and using the already used automated test case it is asserted that the fixes are effective. The testing has been implemented using the IDE Eclipse, JUnit and Selenium.

Taint Analysis

As required by the project specification, the Taint Analysis was performed with the Pixy tool, which tries to detect two major types of web application XSS vulnerabilities and SQL injection.

In this case, the tool has been found only the XSS vulnerabilities. In fact, the following table shows all the vulnerabilities found by the tool, and they have been analysed and described in detail.

Pixy file	Description	Result
xss_changeBio.php_1	file: changeBio.php - line: 22 There is an echo of an array without any tainted variable	False Positive
xss_changePassword.php_1	file: changePassword.php - line: 43 There is an echo of an array without any tainted variable	False Positive
xss_changeUsername.php_1	file: changeUsername.php - line: 23 There is an echo of an array without any tainted variable	False Positive
xss_createBrand.php_1	file: createBrand.php - line: 25 There is an echo of an array without any tainted variable	False Positive
xss_createCategories.php_1	file: createCategories.php - line: 25 There is an echo of an array without any tainted variable	False Positive
xss_createOrder.php_1	file: createOrder.php - line: 70 There is an echo of an array without any tainted variable	False Positive
xss_createProduct.php_1	file: createProduct.php - line: 43 There is an echo of an array without any tainted variable	False Positive
xss_createUser.php_1	file: createUser.php - line: 30	False Positive

	There is an echo of an array without any	
	tainted variable	Falsa Daaitissa
xss_dashboard.php_3	file: dashboard.php - line: 75	False Positive
	The variable is creating counting the rows of a	
	sql table, so the echo can contain only a number	
voc dochboard php 4	file: dashboard.php - line: 87	False Positive
xss_dashboard.php_4	The variable is creating counting the rows of a	raise Positive
	sql table, so the echo can contain only a	
	number	
xss_dashboard.php_5	file: dashboard.php - line: 101	False Positive
x33_ua311b0a1u.p11p_5	The variable is creating counting the rows of a	1 alse i Ositive
	sql table, so the echo can contain only a	
	number	
xss_dashboard.php_10	file: dashboard.php - line: 153	True Positive
7.00_dd01150d14.p11p_10	The username of the logged user is displayed	Truo r contro
	without any sanitization	
xss_dashboard.php_11	file: dashboard.php - line: 154	False Positive
7.00_00.00.00.00.00.pp_	The variable is creating counting the rows of a	
	sql table, so the echo can contain only a	
	number	
xss_editBrand.php_1	file: editBrand.php - line: 25	False Positive
	There is an echo of an array without any	
	tainted variable	
xss_editCategories.php_1	file: editCategories.php - line: 25	False Positive
	There is an echo of an array without any	
	tainted variable	
xss_editOrder.php_1	file: editOrder.php - line: 87	False Positive
	There is an echo of an array without any	
	tainted variable	
xss_editPayment.php_1	file: editPayment.php - line: 31	False Positive
	There is an echo of an array without any	
	tainted variable	
xss_editProduct.php_1	file: editProduct.php - line: 31	False Positive
	There is an echo of an array without any	
	tainted variable	
xss_editProductImage.php_1	file: editProductImage.php - line: 35	False Positive
	There is an echo of an array without any	
	tainted variable	
xss_editUser.php_1	file: editUser.php - line: 27	False Positive
	There is an echo of an array without any	
(115	tainted variable	- ···
xss_fetchBrand.php_1	file: fetchBrand.php - line: 48	True Positive
	There is an echo which displays the brand	
von fotob Cotogorias aba 4	name without any sanitization	True Decition
xss_fetchCategories.php_1	file: fetchCategories.php - line: 48	True Positive
	There is an echo which displays the category	
voo fotobOrdor aba 4	name without any sanitization	True Decitive
xss_fetchOrder.php_1	file: fetchOrder.php - line: 71 There is an echo which displays the client	True Positive
	There is an echo which displays the client	
	contact without any sanitization	

xss_fetchOrderData.php_1	file: fetchOrderData.php - line: 19	False Positive
	There is an echo of an array without any	
	tainted variable	
xss_fetchProduct.php_1	file: fetchProduct.php - line: 83	True Positive
	There is an echo which displays the product	
	name, the brand name and category name	
	without any sanitization	
xss_fetchProductData.php_1	file: fetchProductData.php - line: 12	True Positive
	There is an echo which displays the product	
	name without any sanitization	
xss_fetchProductImageUrl.php_	file: fetchProductImageUrl.php - line: 13	False Positive
1	Product image is selected from the database	
	which is not a potential risk since it is a file	
xss_fetchSelectedBrand.php_1	file: fetchSelectedBrand.php - line: 16	False Positive
	There is an echo of an array without any	
	tainted variable	
xss_fetchSelectedCategories.ph	file: fetchSelectedCategories.php - line: 16	False Positive
p_1	There is an echo of an array without any	T GIOO T CONTO
Ρ_ '	tainted variable	
xss_fetchSelectedProduct.php_	file: fetchSelectedProduct.php - line: 16	False Positive
1	There is an echo of an array without any	1 disc i ositivo
ı	tainted variable	
van fotob Colocted I loor php 1		False Positive
xss_fetchSelectedUser.php_1	file: fetchSelectedUser.php - line: 16	raise Positive
	There is an echo of an array without any	
fatabilia a aba 4	tainted variable	Tour Desition
xss_fetchUser.php_1	file: fetchUser.php - line: 47	True Positive
	There is an echo which display the username	
van sistOudauDauaut ulau 1	without any sanitization	True Positive
xss_getOrderReport.php_1	file: getOrderReport.php - line: 49	True Positive
	There is an echo which display informations	
	without any sanitization	Falsa Davida
xss_index.php_2	file: index.php - line: 100	False Positive
	\$PHP SELF is a variable that returns the	
	current script being executed	
xss_orders.php_6	file: orders.php - line: 37	True Positive
	There is an echo of GET variable `i`, which can	
	be exploited by the user	
xss_orders.php_11	file: orders.php - line: 111	True Positive
	There is an echo which displays the product	
	name without any sanitization	
xss_orders.php_20	file: orders.php - line: 287	False Positive
	The variable contains only objects of Date type	
xss_orders.php_21	file: orders.php - line: 293	True Positive
•	There is an echo which displays the client	
	name without any sanitization	
xss_orders.php_22	file: orders.php - line: 299	True Positive
	There is an echo which displays the client	
	contact without any sanitization	
xss_orders.php_27	file: orders.php - line: 345	True Positive
	There is an echo which displays the product	
	name without any sanitization	
	name without any sanitization	<u> </u>

xss_orders.php_29	file: orders.php - line: 353 Rate is updated through javascript when user	False Positive
	click on the product	
xss_orders.php_31	file: orders.php - line: 354	False Positive
	There is an echo of a hidden variable without	
	vulnerable parameters	
xss_orders.php_32	file: orders.php - line: 365	False Positive
• •	Quantity is displayed only if greater than 0, so	
	an implicit cast is done	
xss_orders.php_35	file: orders.php - line: 380	False Positive
_ ' '-	Quantity is displayed only if greater than 0, so	
	an implicit cast is done	
xss_orders.php_37	file: orders.php - line: 383	False Positive
	The input total can't be manipulated by the	
	user	
xss_orders.php_39	file: orders.php - line: 385	False Positive
7.00_0.00.0.pp_00	There is an echo of a hidden variable without	T GIGG T GOILLYG
	vulnerable parameters	
xss_orders.php_41	file: orders.php - line: 404	False Positive
700_01d010.p11p_11	The input subtotal can't be manipulated by the	T GIOO T CONTVO
	user	
xss_orders.php_42	file: orders.php - line: 405	False Positive
735_01de15.p11p_42	There is an echo of a hidden variable without	i alse i ositive
	vulnerable parameters	
you orders php 42		False Positive
xss_orders.php_43	file: orders.php - line: 412	raise Positive
	The input totalAmount can't be manipulated by the user	
xss_orders.php_44	file: orders.php - line: 413	False Positive
xss_orders.prip_44	There is an echo of a hidden variable without	I alse Fusitive
	vulnerable parameters	
vec orders php 45	file: orders.php - line: 419	True Positive
xss_orders.php_45	· ·	True Positive
vac ordere php 46	The input discount is editable by the user	False Positive
xss_orders.php_46	file: orders.php - line: 425	raise Positive
	The input grandTotal can't be manipulated by	
voc andone also 47	the user	Folos Positivo
xss_orders.php_47	file: orders.php - line: 426	False Positive
	There is an echo of a hidden variable without	
	vulnerable parameters	Falsa Dasitina
xss_orders.php_50	file: orders.php - line: 432	False Positive
	The input vat can't be manipulated by the user	E I D '0'
xss_orders.php_51	file: orders.php - line: 433	False Positive
	There is an echo of a hidden variable without	
	vulnerable parameters	
xss_orders.php_52	file: orders.php - line: 439	True Positive
	The input gstn is editable by the user	
xss_orders.php_53	file: orders.php - line: 448	True Positive
	The input paid is editable by the user	
xss_orders.php_54	file: orders.php - line: 454	False Positive
	The input due can't be manipulated by the user	
xss_orders.php_55	file: orders.php - line: 455	False Positive

	There is an echo of a hidden variable without	
	vulnerable parameters	– – •••
xss_orders.php_64	file: orders.php - line: 513	True Positive
	There is an echo of GET variable `i`, which can	
	be exploited by the user	T D ''.'
xss_printOrder.php_1	file: printOrder.php - line: 193	True Positive
	There is an echo which displays the client and	
	contact name and product name without any	
	sanitization	
xss_product.php_1	file: product.php - line: 109	True Positive
	There is an echo which displays the brand	
	name without any sanitization	
xss_product.php_2	file: product.php - line: 128	True Positive
	There is an echo which displays the category	
	name without any sanitization	
xss_product.php_3	file: product.php - line: 267	True Positive
	There is an echo which displays the brand	
	name without any sanitization	
xss_product.php_4	file: product.php - line: 286	True Positive
	There is an echo which displays the category	
	name without any sanitization	
xss_removeBrand.php_1	file: removeBrand.php - line: 24	False Positive
_ ' ' ' -	There is an echo of an array without any	
	tainted variable	
xss_removeCategories.php_1	file: removeCategories.php - line: 24	False Positive
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	There is an echo of an array without any	
	tainted variable	
xss_removeOrder.php_1	file: removeOrder.php - line: 26	False Positive
700_10111010001401.pnp_1	There is an echo of an array without any	1 000 1 0011110
	tainted variable	
xss_removeProduct.php_1	file: removeProduct.php - line: 24	False Positive
x33_remover roddet.prip_1	There is an echo of an array without any	1 disc i ositivo
	tainted variable	
xss_removeUser.php_1	file: removeUser.php - line: 24	False Positive
xss_removeOser.pnp_r		I dise Fusitive
	There is an echo of an array without any tainted variable	
vac actting php 1		True Positive
xss_setting.php_1	file: setting.php - line: 35	True Positive
	There is an echo of the username which can	
	be exploited by the user	False Positive
xss_setting.php_2	file: setting.php - line: 41	Faise Positive
	There is an echo of an id variable which cannot	
ven notting the O	be modified by the attacker	True Desities
xss_setting.php_3	file: setting.php - line: 57	True Positive
	There is an echo of the user bio which can be	
	exploited by the user	<u> </u>
xss_setting.php_4	file: setting.php - line: 63	False Positive
	There is an echo of an id variable which cannot	
	be modified by the attacker	
xss_setting.php_5	file: setting.php - line: 99	False Positive
	There is an echo of an id variable which cannot	
	be modified by the attacker	

Fixes

echo ison encode(\$data):

This chapter will list the fixes added to the code to remove XSS vulnerabilities.

```
dashboard.php
Vulnerability:
echo $_SESSION['username'];
Fix:
echo htmlentities($_SESSION['username']);
Vulnerability:
echo ($orderResult['username]']);
Fix:
echo (htmlentities($orderResult['username]']));
fetchBrand.php
Vulnerability:
$output['data'][] = array($row[1]), $activeBrands, $button);
Fix:
$output['data'][] = array(htmlentities($row[1])), $activeBrands, $button);
fetchCategories.php
Vulnerability:
$output['data'][] = array($row[1]), $activeCategories, $button);
Fix:
$output['data'][] = array(htmlentities($row[1])), $activeCategories, $button);
fetchOrder.php
Vulnerability:
$output['data'][] = array($x, $row[1]), $row[2], $row[3], $itemCountRow, $paymentStatus,
$button):
Fix:
$output['data'][] = array($x, $row[1]), htmlentities($row[2]), htmlentities($row[3]),
$itemCountRow, $paymentStatus, $button);
fetchProduct.php
Vulnerability:
$output['data'][] = array($productImage, $row[1], $row[6], $row[5], $brand, $category,
$active,$button);
Fix:
$output['data'][] = array($productImage, htmlentities($row[1]), floatval($row[6]),
$row[5], htmlentities($brand), htmlentities($category), $active,$button);
fetchProductData.php
Vulnerability:
echo ison encode($data);
foreach($data as $datum){$datum[1] = htmlentities($datum[1])}
```

```
fetchUser.php
Vulnerability:
$output['data'][] = array($username, $button);
Fix:
$output['data'][] = array(htmlentities($username), $button);
orders.php
Vulnerability:
echo "Edit Order " . $_GET['i'];
Fix:
echo "Edit Order " . intval($_GET['i']);
Vulnerability:
id='changeProduct".$row['product_id']."'>".$row['product_name']."</option>";id='changePro
duct".$row['product id']."' ".$selected." >"
.($row['product name'])."</option>";
Fix:
id='changeProduct".$row['product_id']."'>".htmlentities($row['product_name'])."</option>";
Vulnerability:
<input type="text" class="form-control" id="orderDate" name="orderDate"</pre>
autocomplete="off" value="<?php echo $data[1] ?>" />
<input type="text" class="form-control" id="clientName" name="clientName"</pre>
placeholder="Client Name" autocomplete="off" value="<?php echo $data[2] ?>" />
<input type="text" class="form-control" id="clientContact" name="clientContact"</pre>
placeholder="Contact Number" autocomplete="off" value="<?php echo $data[3] ?>" />
Fix:
<input type="text" class="form-control" id="orderDate" name="orderDate"</pre>
autocomplete="off" value="<?php echo htmlentities($data[1]) ?>" />
<input type="text" class="form-control" id="clientName" name="clientName"</pre>
placeholder="Client Name" autocomplete="off" value="<?php echo htmlentities($data[2])
?>"/>
<input type="text" class="form-control" id="clientContact" name="clientContact"</pre>
placeholder="Contact Number" autocomplete="off" value="<?php echo
htmlentities($data[3]) ?>" />
Vulnerability:
echo "<option value="".$row['product id']."' id='changeProduct".$row['product id']."'
".$selected." >".$row['product_name']."</option>";
Fix:
echo "<option value="".$row['product_id']."' id='changeProduct".$row['product_id']."'
".$selected." >".htmlentities($row['product_name'])."</option>":
Vulnerability:
<input type="hidden" name="orderld" id="orderld" value="<?php echo $ GET['i']; ?>" />
<input type="hidden" name="orderId" id="orderId" value="<?php echo intval($_GET['i']);</pre>
?>"/>
```

```
printOrder.php
Vulnerability:
$clientName = $orderData[1];
$clientContact = $orderData[2];
$gstn = $orderData[11];
Fix:
$clientName = htmlentities($orderData[1]);
$clientContact = htmlentities($orderData[2]);
$gstn = htmlentities($orderData[11]);
product.php
Vulnerability:
echo "<option value="".$row[0]."'>". $row[1]."</option>";
echo "<option value="".$row[0]."'>". htmlentities($row[1])."</option>";
Vulnerability:
echo "<option value="".$row[0]."'>".$row[1]."</option>";
echo "<option value="".$row[0]."'>".htmlentities($row[1])."</option>";
Vulnerability:
echo "<option value="".$row[0]."'>".$row[1]."</option>";
echo "<option value="".$row[0]."'>".htmlentities($row[1])."</option>";
Vulnerability:
echo "<option value="".$row[0]."'>".$row[1]."</option>";
echo "<option value="".$row[0]."'>".htmlentities($row[1])."</option>";
setting.php
Vulnerability:
<input type="text" class="form-control" id="username" name="username"
placeholder="Usename" value="<?php echo ($result['username']); ?>"/>
Fix:
<input type="text" class="form-control" id="username" name="username"
placeholder="Usename" value="<?php echo (htmlentities($result['username'])); ?>"/>
Vulnerability:
<input type="text" class="form-control" id="bio" name="bio" placeholder="Bio"
value="<?php echo ($result['bio']); ?>"/>
Fix:
<input type="text" class="form-control" id="bio" name="bio" placeholder="Bio"
value="<?php echo (htmlentities($result['bio'])); ?>"/>
```

Conclusion

This analysis allowed to discover 76 vulnerabilities of which only 24 were true positive and then fixed. It is right to remember that this analysis found only the xss vulnerabilities because this was the purpose of the project and that a more detailed analysis could have found other internal issues.