Check Point Report

Vulnerable Application: BodgeIt

Team Name: Group X

Team Members: X, X, X

Vulnerability	Details of the Attack	Screenshot evidence
(C1) **Broken Authentication** since “password” is one of the most commonly used passwords	Attack: **Brute force dictionary attack** 1. Navigate to */bodgeit/login.jsp* and input the username "*test@thebodgeitstore.com*" 2. Input the password "*password*" and click the "*Login*" button.	* included below table
(C2) **SQL Injection**	Attack: **SQL injection** 1) Navigate to /bodgeit/login.jsp 2) Enter *user1@thebodgeitstore.com* into the username field and enter *‘ or ‘1'=’1* into the password field and press the *Login* button.	<<a screenshot showing the result of the attack>> * included below table
(C3) **SQL Injection**	Attack: **SQL Injection** 1) Navigate to */bodgeit/login.jsp* 2) Enter *admin@thebodgeitstore.com' or '1'='1* into the username field and press the *Login* button.	* included below table
(C4) **Insecure Direct Object References** since an attacker can change a parameter value that directly	Attack: **Insecure direct object reference attack** 1) Inspect source code of */bodgeit/login.jsp* to look for clues	* included below table

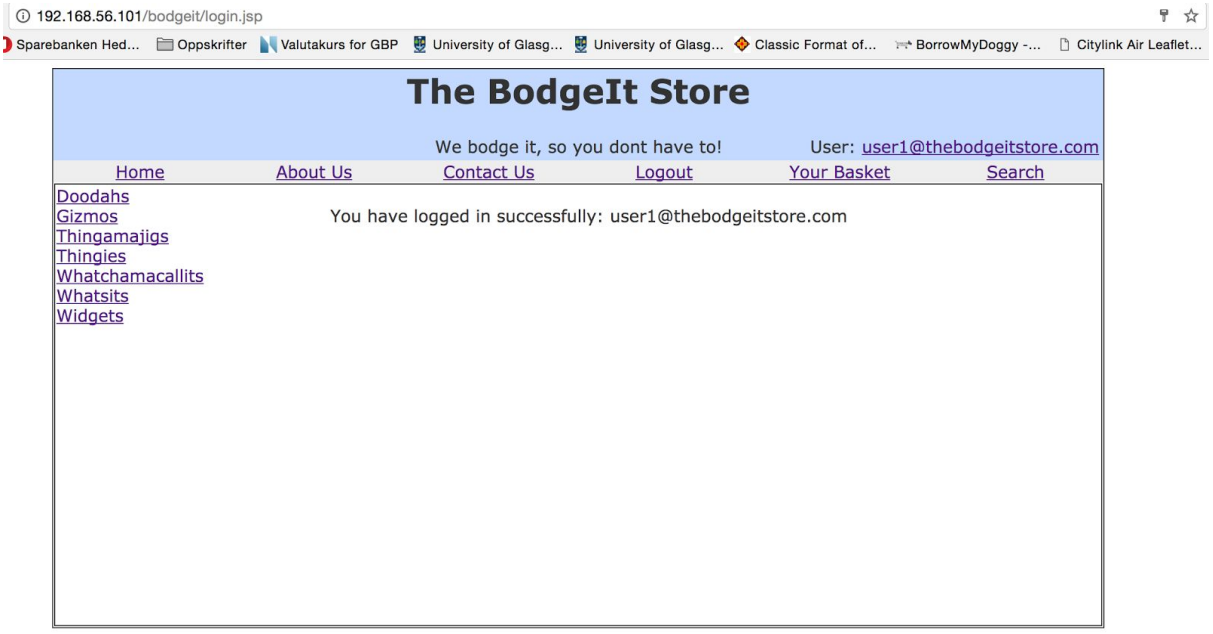
refers to a system object - in this case the admin page.	2) Upon finding the line: *<!-- td align="center" width="16%">Admin</td-->* Navigate to */bodgeit/admin.jsp* 3) You have now found an insecure admin page for the Bodgeit Store Website	
(C5) **Missing Function Level Access Control** since the user can change the URL to enable access the debug function	Attack: **Forced browsing attack** 1) Navigate to */bodgeit/basket.jsp*. Append *?debug=true* to the url to create */bodgeit/basket.jsp?debug=true* 2) Create an exception by using Firebug (or any browser add-on to tamper data) to change the value of the "*quantity*" parameter from a numerical value to an alphabetic value such as "*a*". 3) Upon updating the basket, the response will include diagnostic data for the exception that was created.	* included below table
(C6) **Cross-Site Scripting (XSS)**	Attack: **XSS attack** 1) Navigate to */bodgeit/search.jsp* 2) Inject *<script>alert("XSS")</script>* into the textbox and press the "*search*" button 3) A popup will be generated with the text "*XSS*"	* included below table
(C7) **Cross-Site Scripting (XSS)**	Attack: **XSS attack** 1) Navigate to */bodgeit/register.jsp* and enter *john@doe.com<script>alert("XSS")</script>* into the form. Press *submit*. 2) Since the email is displayed at the top-right corner of the page once you are logged in, the XSS script will also be run and so a popup with the text "*XSS*" appears	* included below table

(C9) **Broken Authentication**	Attack: **Cookie manipulation** 1) As a guest user, add an item to your basket in order to generate the *b_id* cookie. 2) Upon viewing the admin.jsp panel, you can see the *b_id*'s of other users. Edit your new *b_id* cookie and change the value from the current value to *1*. 3) Go to your basket and update the basket then refresh the page. You will now be viewing the test user's basket	* included below table
(C12) **Injection**	Attack: **URL string injection** 1) Login as any user. I used user: *test@thebodgeitstore.com* and pass: *password* 2) Navigate to the password-changing page by clicking on the user's email at the top-right corner of the page. Enter a password. 3) Inspect the form element and find the following line: *<form method="POST">* 4) Change "*POST*" to "*GET*" and click the "*Submit*" button	* included below table
Cross Site Request Forgery	1) Grab the GET request for password, e.g *http://0.0.0.0/bodgeit/password.jsp?password1=qwerty&password2=qwerty* 2) Execute CSRF by submitting the following as feedback in the contact form: *<img src='http://0.0.0.0/bodgeit/password.jsp?password1=qwerty&password2=qwerty'>* 3) When the Admin checks the comments in the contact.jsp, the "*src*" parameter value sends a request to change the admin password	* included below table* included below table

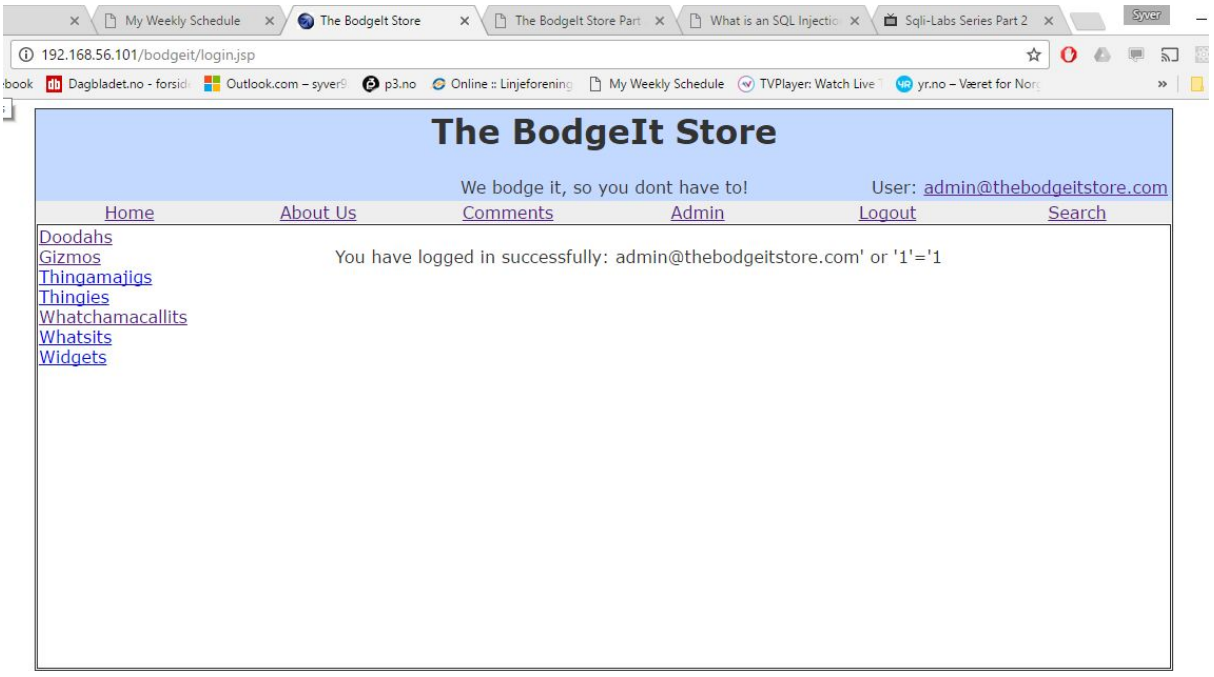
Screenshots:

(C1) Insecure authentication

(C2) SQL Injection



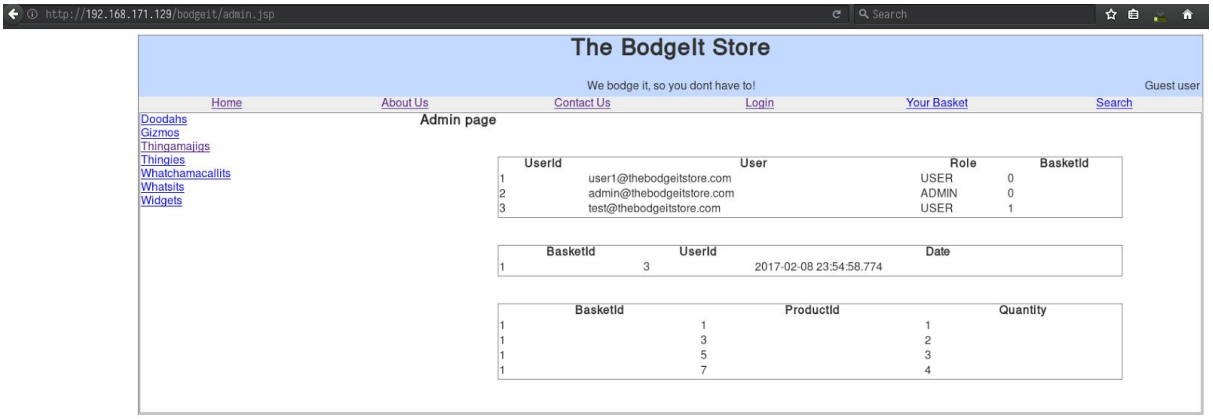
(C3) SQL Injection



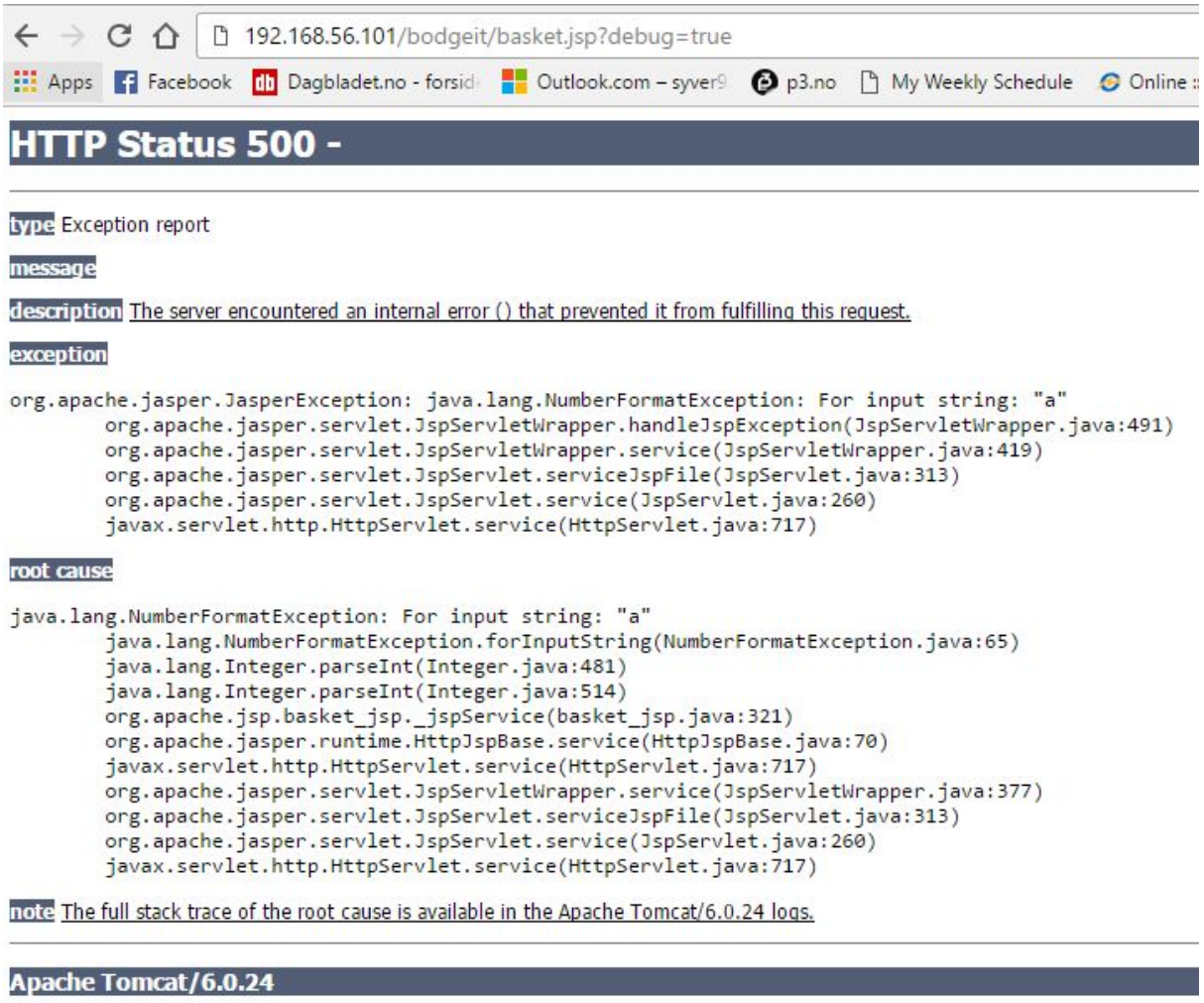
(C4) Insecure Direct Object References

view-source:http://192.168.171.129/bodgeit/

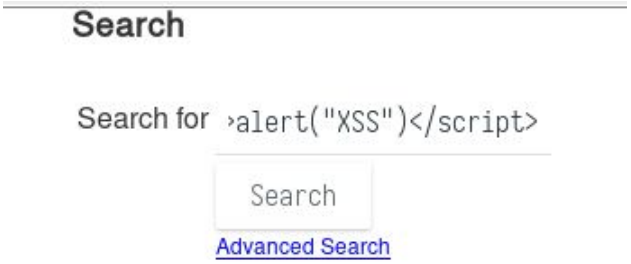
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
<title>The BodgeIt Store</title>
<link href="style.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="./js/util.js"></script>
</head>
<body>

<center>
<table width="80%" class="border">
<tr BGCOLOR=#C3D9FF>
<td align="center" colspan="6">
<H1>The BodgeIt Store</H1>
<table width="100%" class=\"noborder\">
<tr BGCOLOR=#C3D9FF>
<td align="center" width="30%"> </td>
<td align="center" width="40%">We bodge it, so you dont have to!</td>
<td align="center" width="30%" style="text-align: right" >
Guest user

</tr>
</table>
</td>
</tr>
<tr>
<td align="center" width="16%" BGCOLOR=#EEEEEE><a href="home.jsp">Home</a></td>
<td align="center" width="16%" BGCOLOR=#EEEEEE><a href="about.jsp">About Us</a></td>

<td align="center" width="16%" BGCOLOR=#EEEEEE><a href="contact.jsp">Contact Us</a></td>
<!-- td align="center" width="16%"><a href="admin.jsp">Admin</a></td-->

<td align="center" width="16%" BGCOLOR=#EEEEEE>

        <a href="login.jsp">Login</a>

</td>

<td align="center" width="16%" BGCOLOR=#EEEEEE><a href="basket.jsp">Your Basket</a></td>

<td align="center" width="16%" BGCOLOR=#EEEEEE><a href="search.jsp">Search</a></td>
</tr>
```



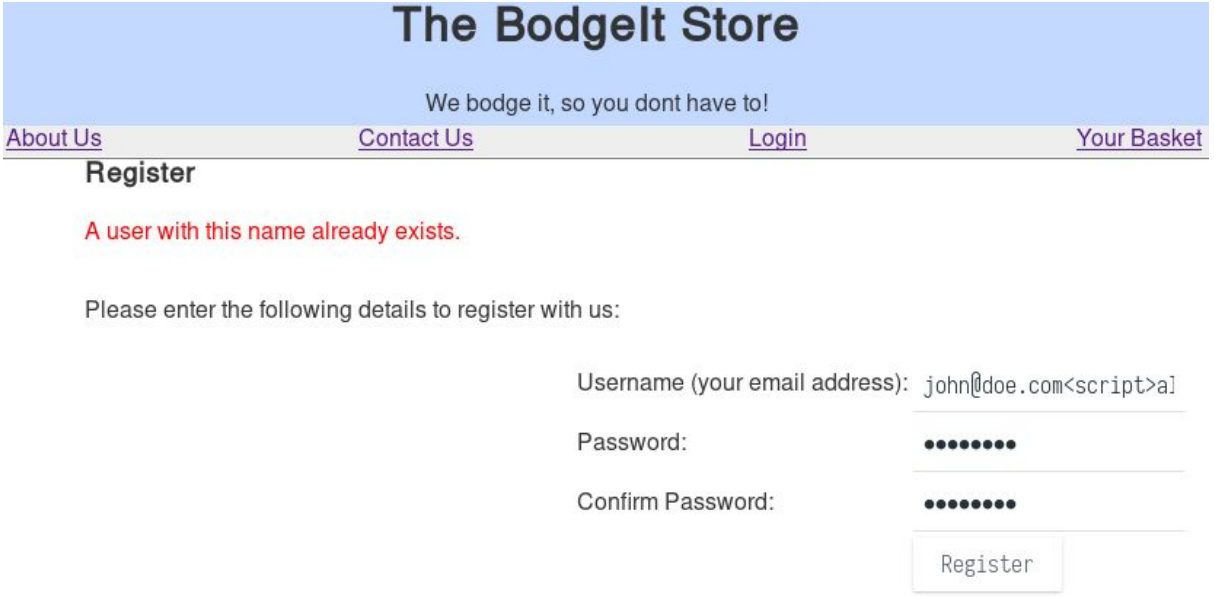
(C5) Missing Function Level Access Control

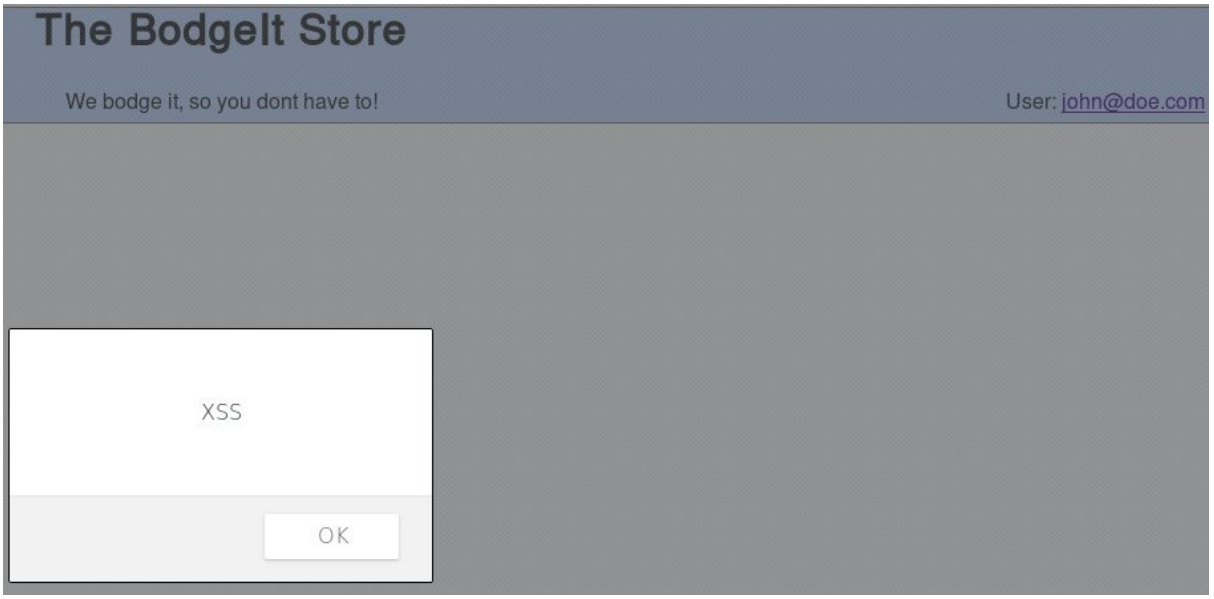


(C6) Cross-Site Scripting (XSS)

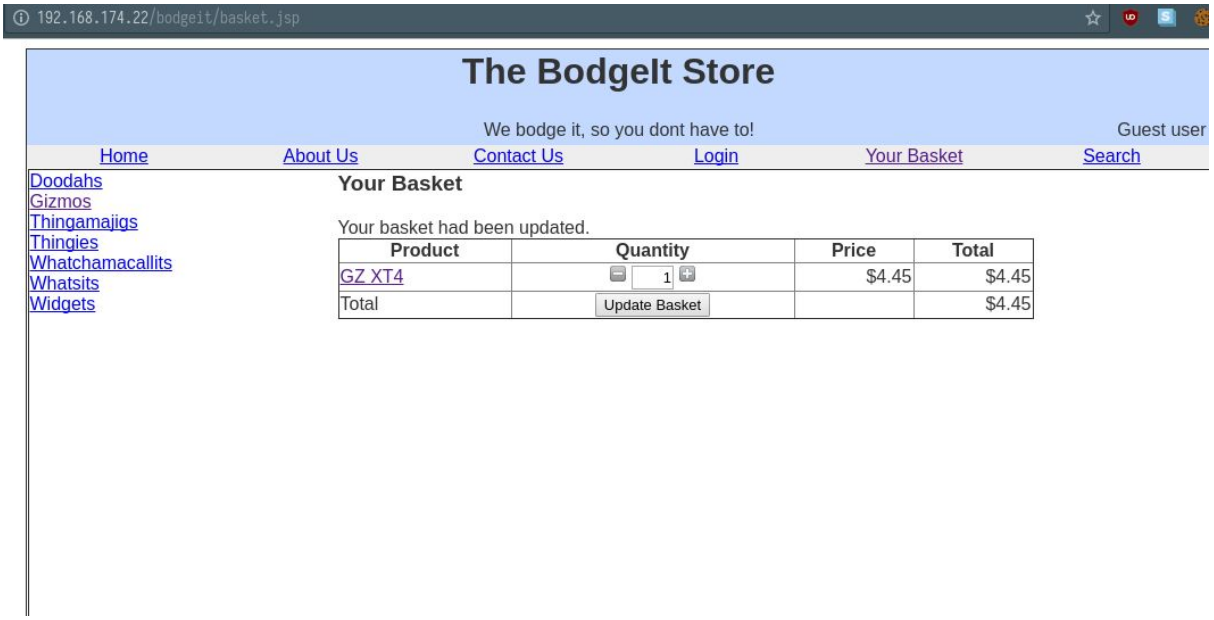


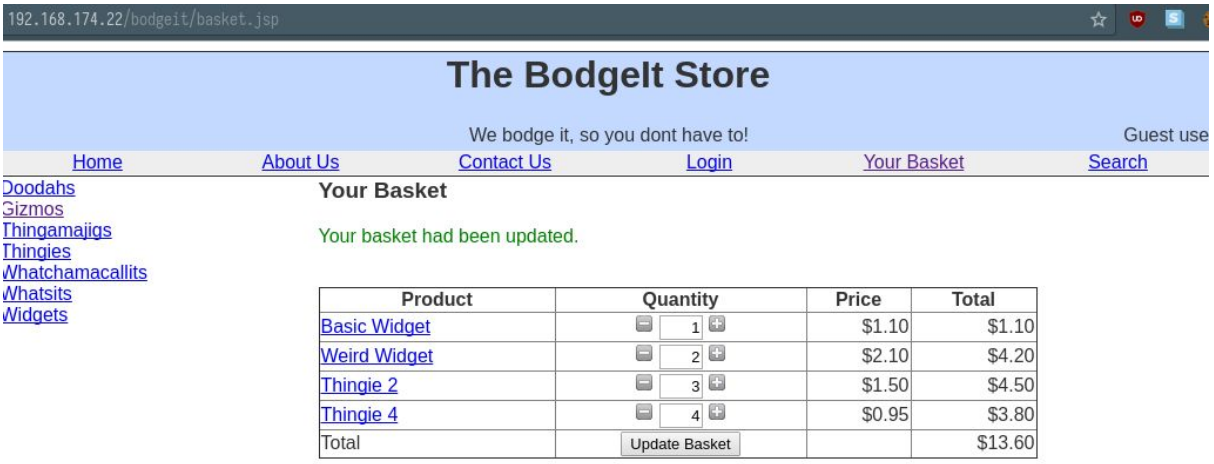
(C7) Cross-Site Scripting (XSS)



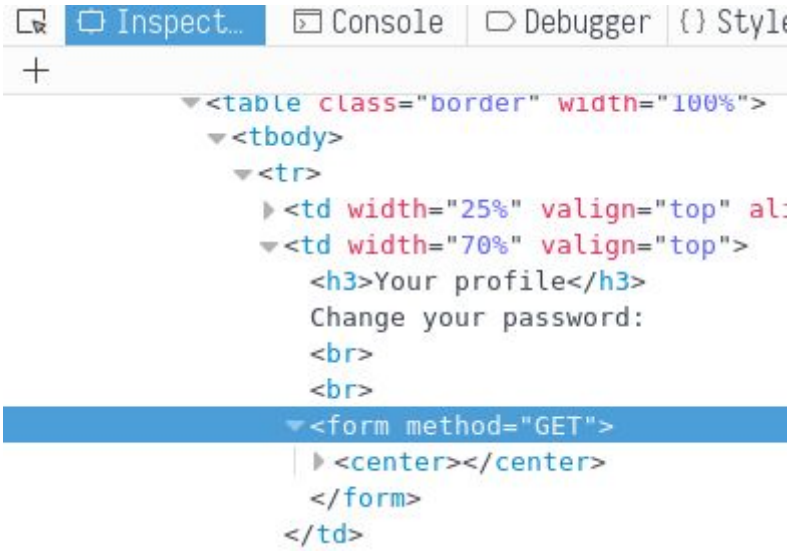


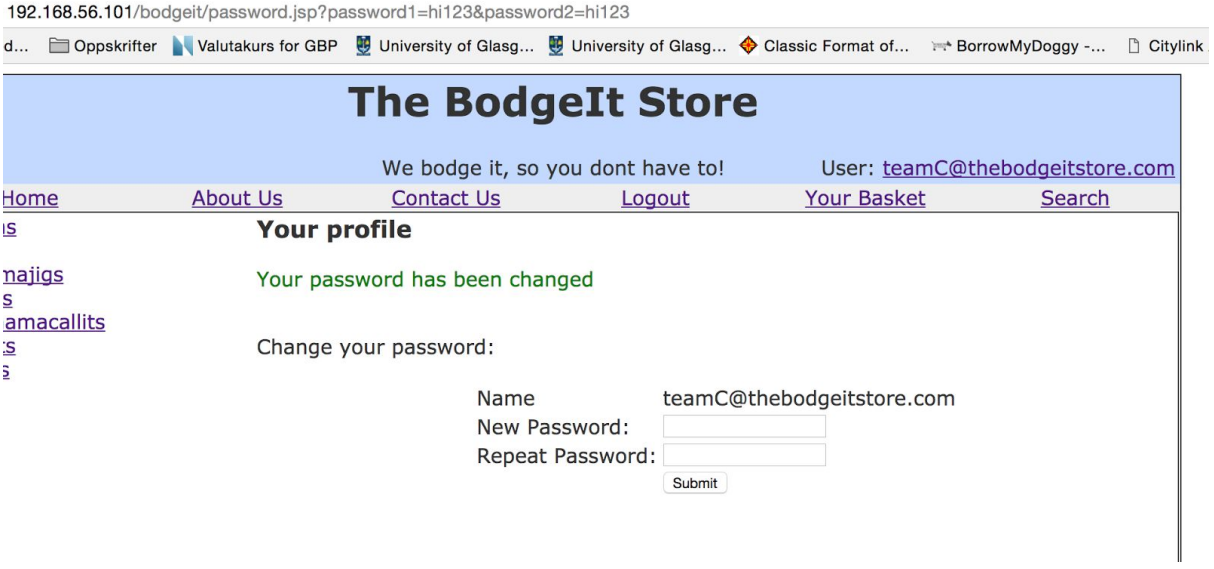
(C9) Broken Authentication



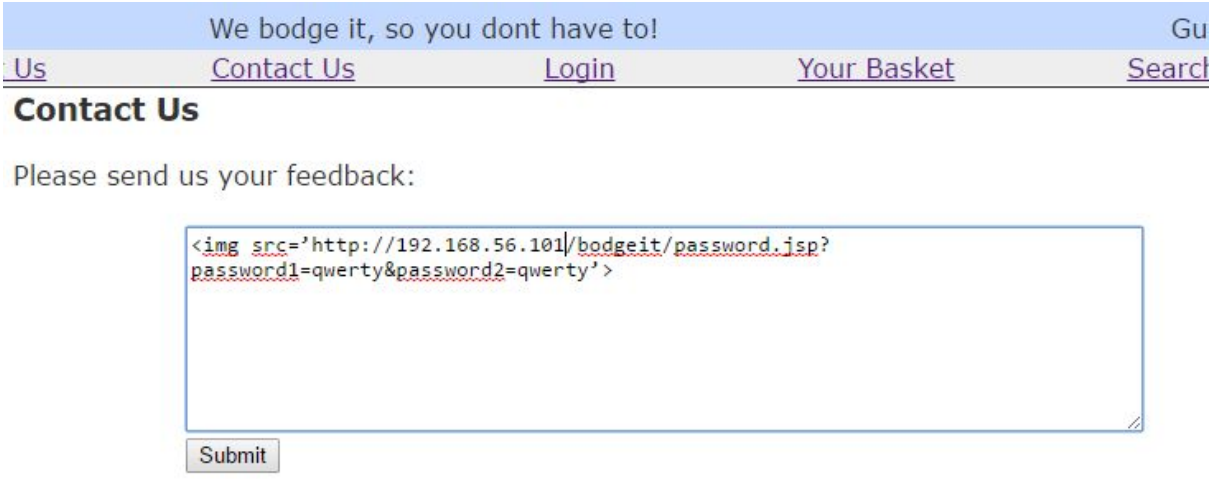


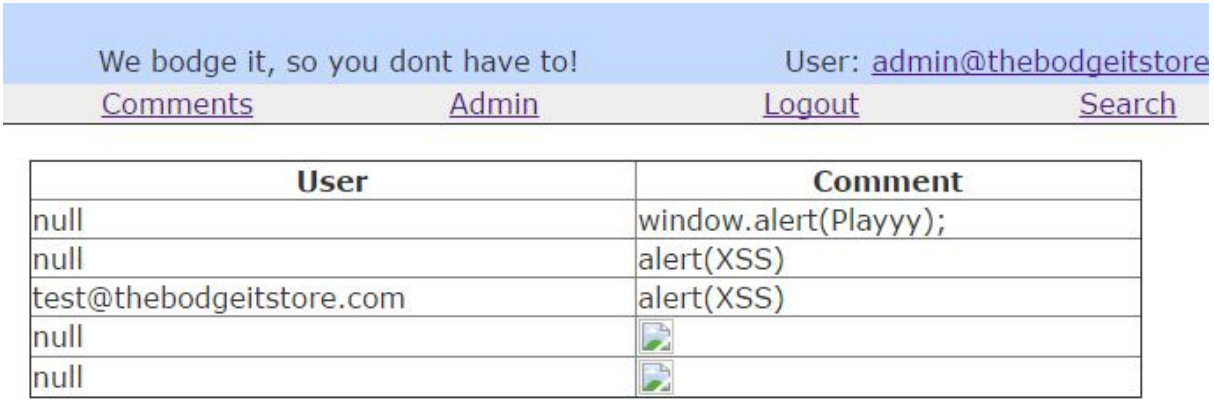
(C12) Injection





Cross Site Request Forgery





User	Comment
null	window.alert(Playyy);
null	alert(XSS)
test@thebodgeitstore.com	alert(XSS)
null	
null	