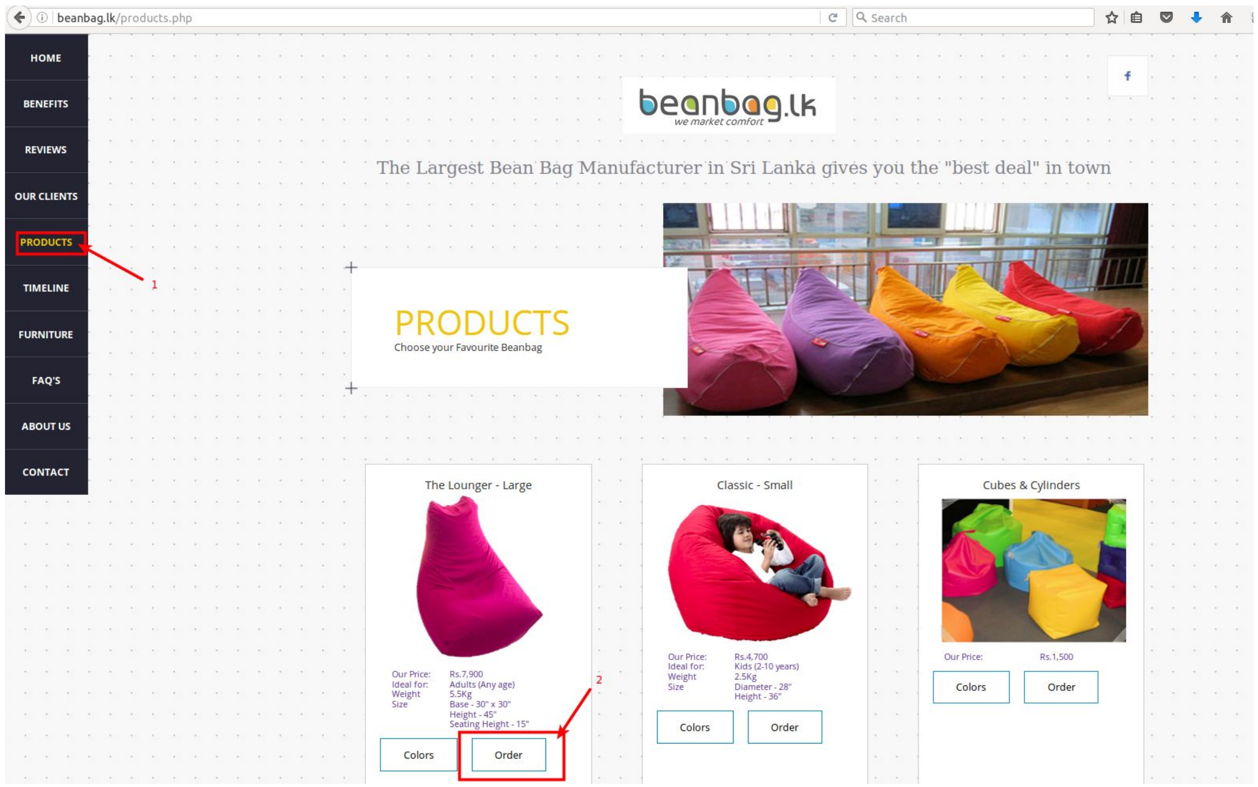
# Query String Injection and Cross Site Scripting (XSS) Vulnerability in BeanBag.LK website

The http://beanbag.lk website is identified to be vulnerable to the above mentioned attacks and this report provides detailed instructions on exploiting the vulnerabilities, root cause for the issue and the steps for fixing the source code for preventing the detected vulnerabilities.

## Exploit

In the Products page of BeanBag.LK website http://beanbag.lk/products.php , each item listed has a button for ordering.



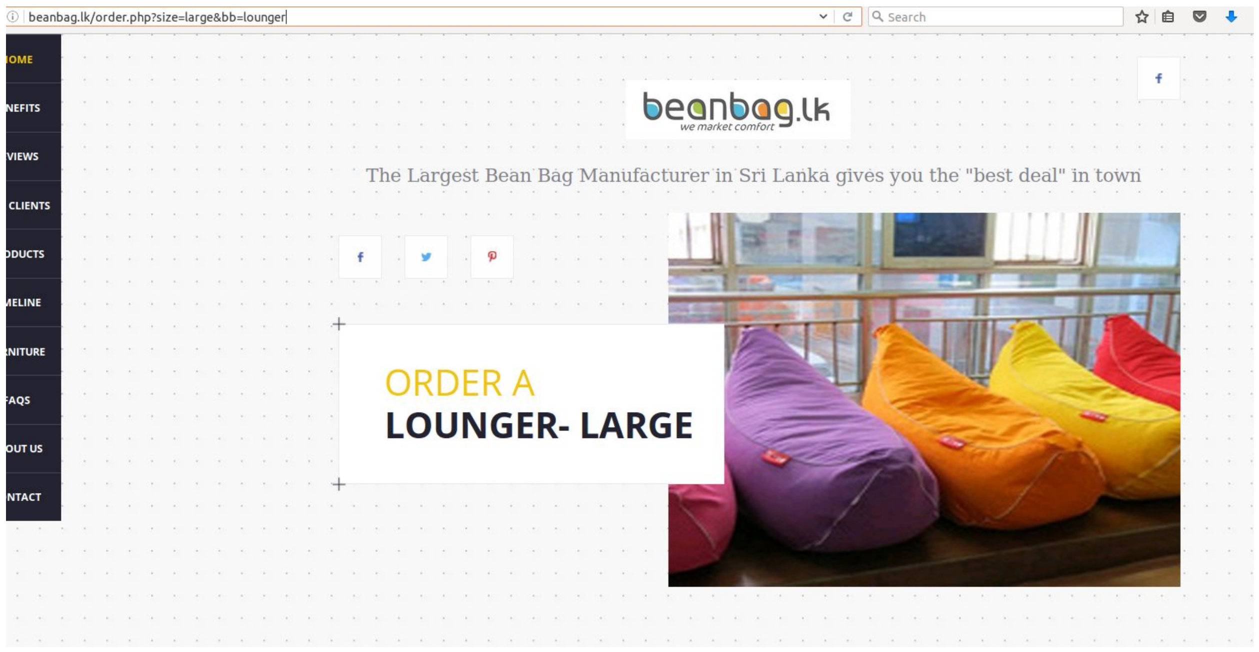
The Lounger - Large
Our Price: Rs.7,900
Ideal for: Adults (Any age)
Weight 5.5Kg
Size Base - 30" x 30"
Height - 45"
Seating Height - 15"

Classic - Small
Our Price: Rs.4,700
Ideal for: Kids (2-10 years)
Weight 2.5Kg
Size Diameter - 28"
Height - 36"

Cubes & Cylinders
Our Price: Rs.1,500

When we click this ‘Order’ button, it redirects the browser to the order.php web page where in the URL, it sends two query parameters ‘size’ and ‘bb’.

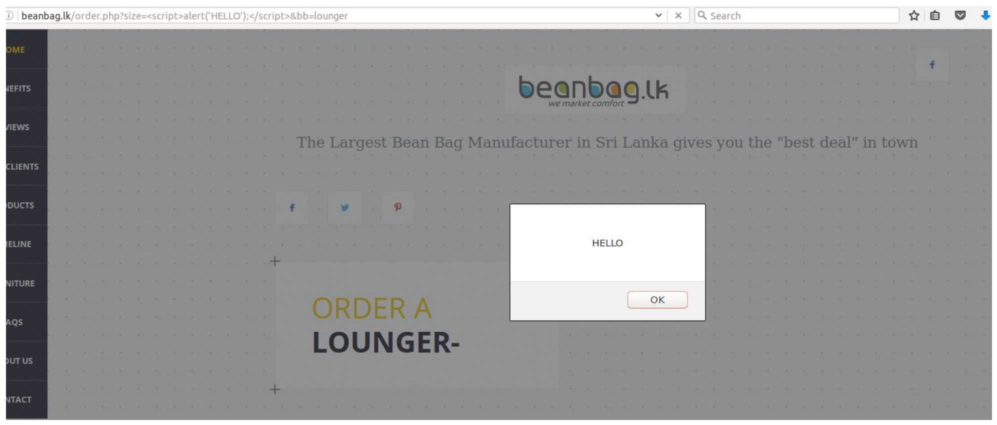
http://beanbag.lk/order.php?size=large&bb=lounger


beanbag.lk/order.php?size=large&bb=lounger
Search
OME
NEFITS
VIEWS
CLIENTS
DUCTS
MELINE
NITURE
FAQS
OUT US
NTACT
beanbag.lk
we market comfort
The Largest Bean Bag Manufacturer in Sri Lanka gives you the "best deal" in town
ORDER A
LOUNGER- LARGE

Let’s inject a javascript to the ‘size’ query parameter to check if XSS is possible.

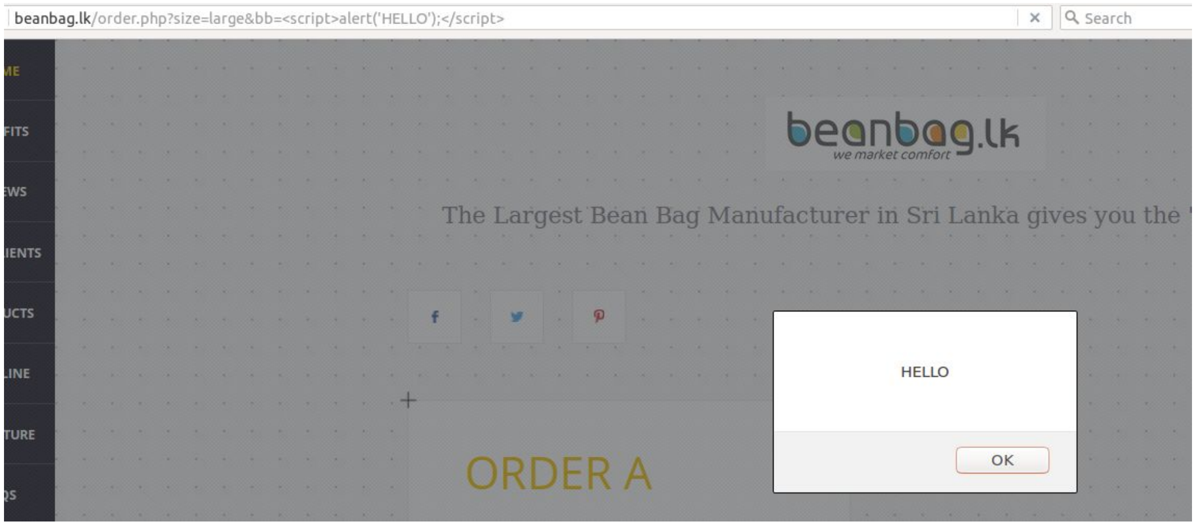
http://beanbag.lk/order.php?size=<script>alert('HELLO');</script>&bb=lounger

We can see that XSS is possible.


beanbag.lk/order.php?size=<script>alert('HELLO');</script>&bb=lounger
Search
OME
EFITS
IEWS
CLIENTS
DUCTS
ELINE
NITURE
AQS
OUT US
NTACT
beanbag.lk
we market comfort
The Largest Bean Bag Manufacturer in Sri Lanka gives you the "best deal" in town
HELLO
OK
ORDER A
LOUNGER-

Let's inject a javascript to the 'bb' query parameter for checking if that is also vulnerable to XSS.

http://beanbag.lk/order.php?size=large&bb=<script>alert('HELLO');</script>


beanbag.lk/order.php?size=large&bb=<script>alert('HELLO');</script>

The Largest Bean Bag Manufacturer in Sri Lanka gives you the

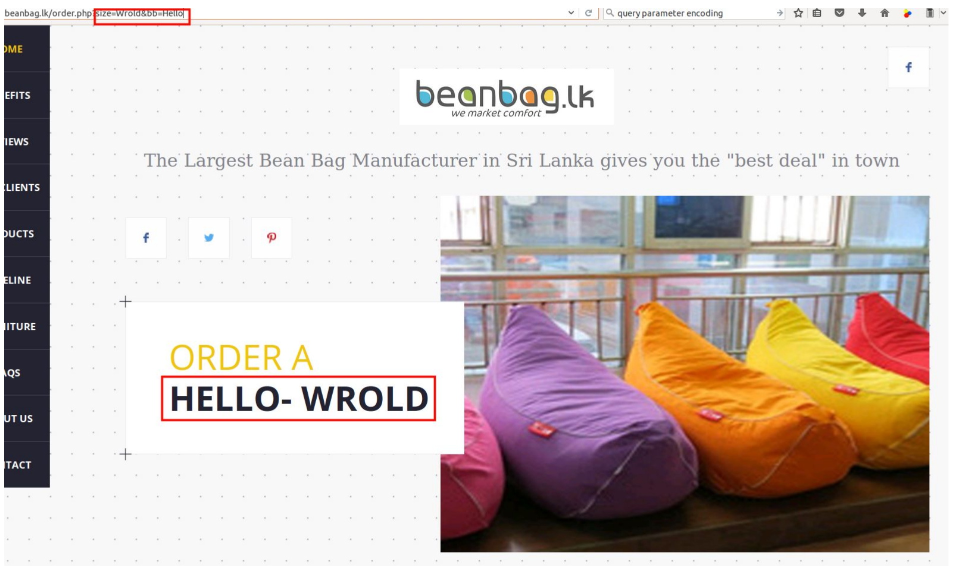
ORDER A

HELLO

We can see that 'bb' query parameter is also vulnerable to XSS.

Now let's just change the values of the 'size' and 'bb' query parameters and see what happens. Here I give the value for 'size' as 'World' and for 'bb' I give the value 'Hello'.

http://beanbag.lk/order.php?size=Wrold&bb=Hello


beanbag.lk/order.php?size=Wrold&bb=Hello

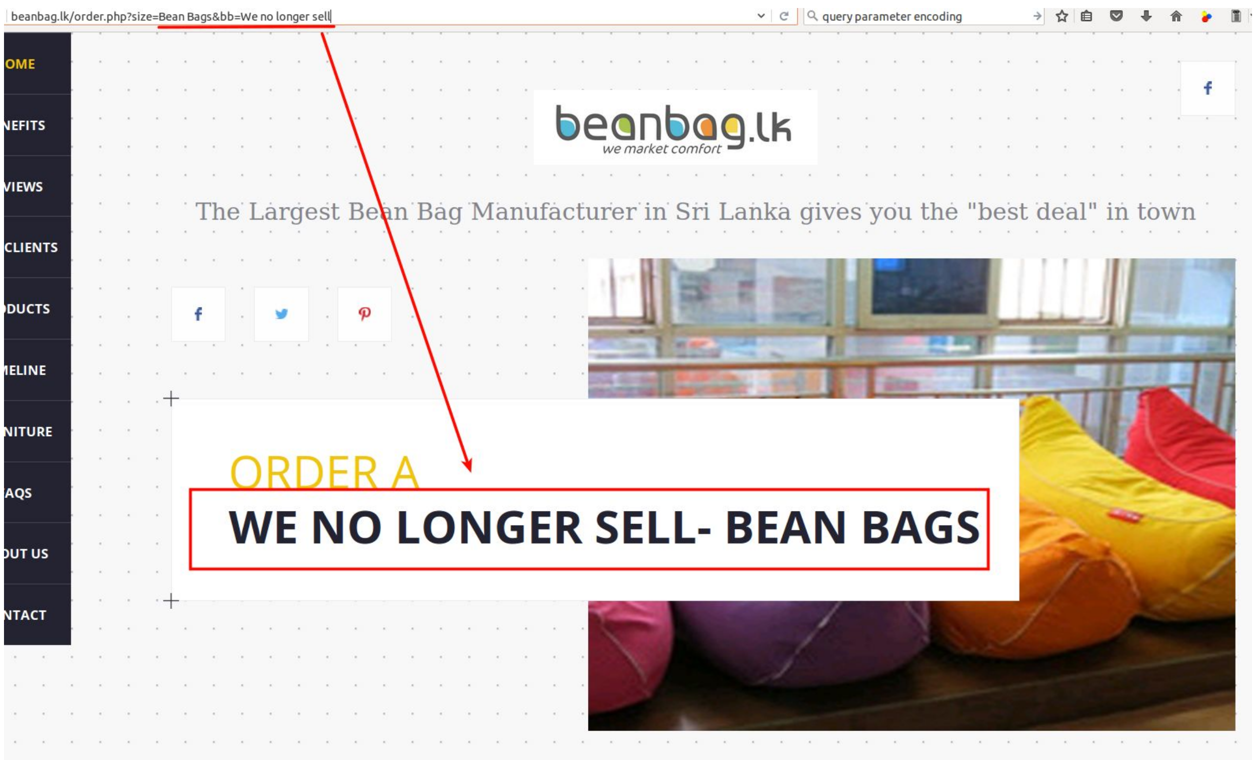
The Largest Bean Bag Manufacturer in Sri Lanka gives you the "best deal" in town

ORDER A
HELLO- WROLD

The values I entered for the query parameters are nicely displayed in the web page.
What if someone crafts a URL like below and shares publicly ?

http://beanbag.lk/order.php?size=Bean Bags&bb=We no longer sell

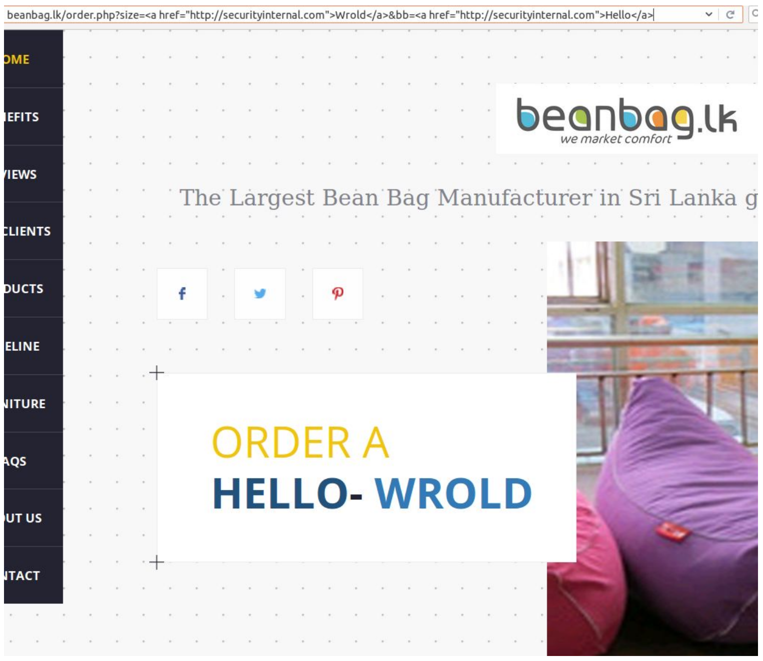
To hide the content of the URL, the URL can be shortened using a URL shortener before sharing, so that the person who clicks the URL would not notice anything suspicious at the first look.


beanbag.lk/order.php?size=Bean Bags&bb=We no longer sell
query parameter encoding
OME
NEFITS
VIEWS
CLIENTS
DUCTS
IELINE
NITURE
AQS
UT US
NTACT
beanbag.lk
we market comfort
The Largest Bean Bag Manufacturer in Sri Lanka gives you the "best deal" in town
ORDER A
WE NO LONGER SELL- BEAN BAGS

This is not good for the good name of the business :) !!!

The attacker can even add hyperlinks to the web page and try to redirect the end user to a different website.

http://beanbag.lk/order.php?size=<a href="http://securityinternal.com">Wrold</a>&bb=<a href="http://securityinternal.com">Hello</a>

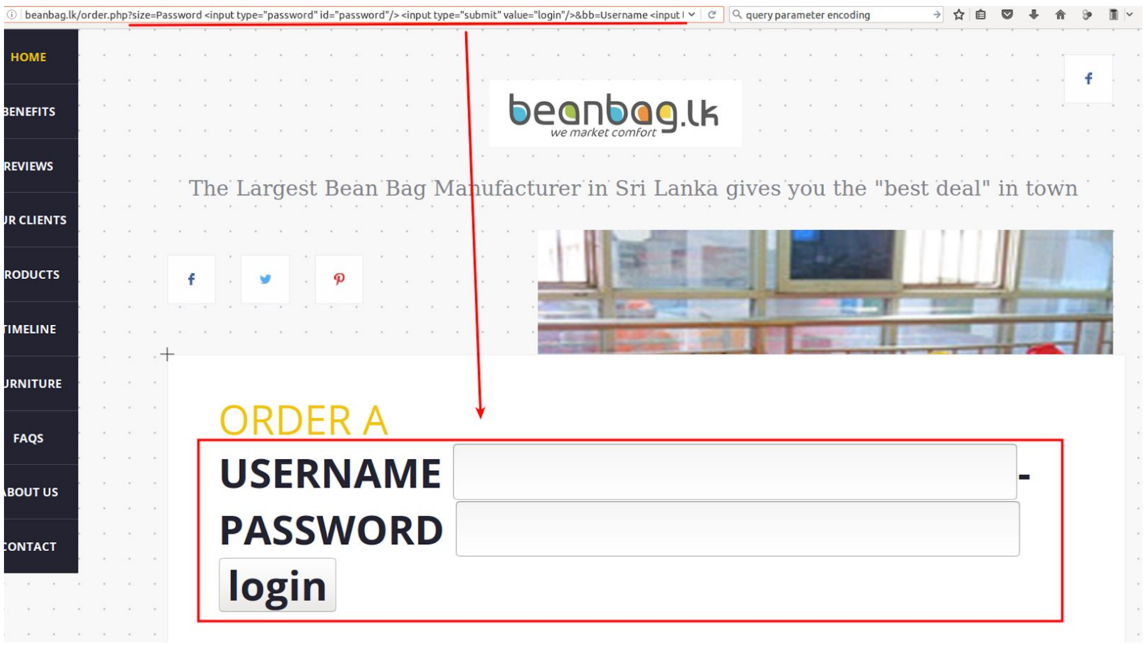

beanbag.lk/order.php?size=<a href="http://securityinternal.com">Wrold</a>&bb=<a href="http://securityinternal.com">Hello</a>

The Largest Bean Bag Manufacturer in Sri Lanka g

ORDER A
HELLO- WROLD

The attacker can even inject HTML text boxes like username password to the webpage and try to steal some credentials of the end user.

http://beanbag.lk/order.php?size=Password <input type="password" id="password"/> <input type="submit" value="login"/>&bb=Username <input type="username" id="username"/>


beanbag.lk/order.php?size=Password <input type="password" id="password"/> <input type="submit" value="login"/>&bb=Username <input t

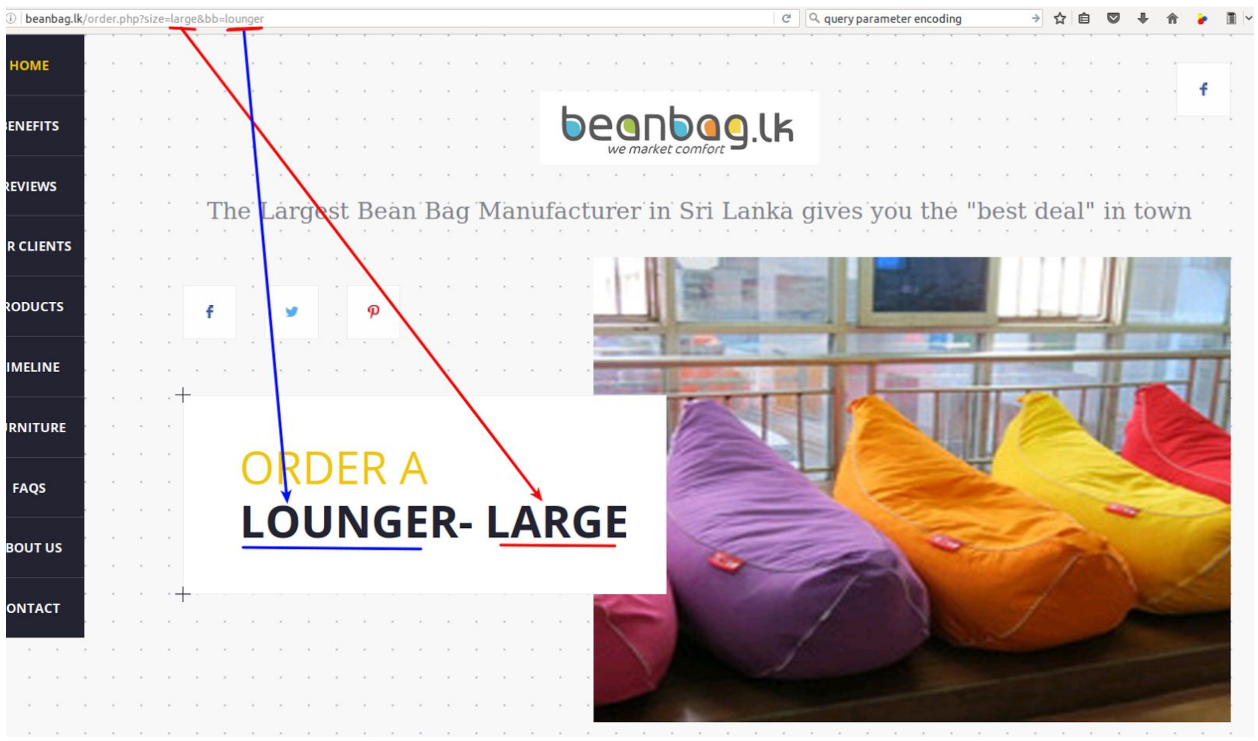
The Largest Bean Bag Manufacturer in Sri Lanka gives you the "best deal" in town

ORDER A
USERNAME -
PASSWORD
login

## Root Cause

The reason for the Cross Site Scripting vulnerability is directly displaying the values of ‘size’ and ‘bb’ query parameters in the order.php web page, without proper output encoding (sanitization).

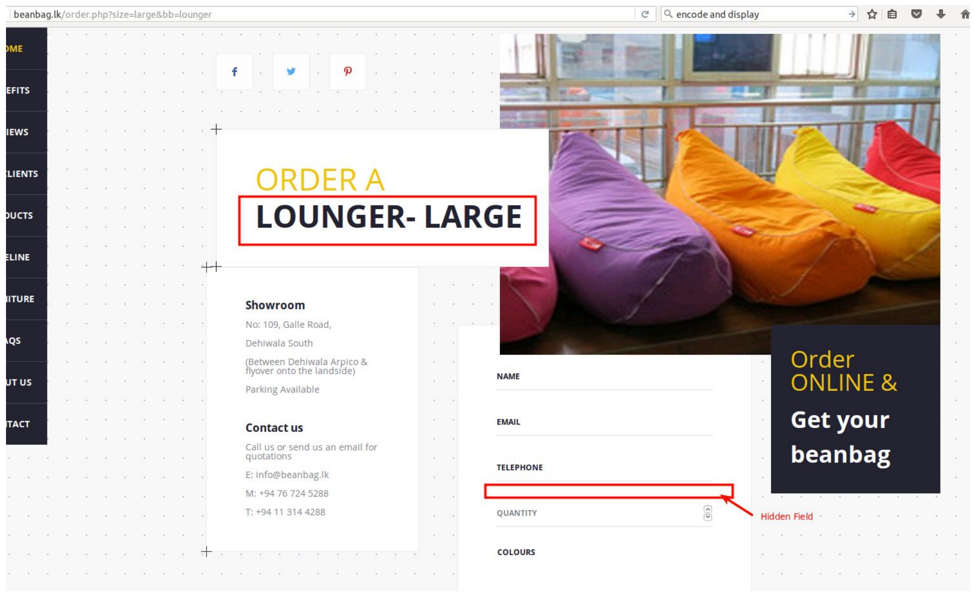
http://beanbag.lk/order.php?size=large&bb=lounger



If the values of the query parameters are properly sanitized (escaped), then executing malicious scripts in the user’s browser would not be possible.

The other vulnerability is the Query String Injection where an attacker can simply modify the query parameter values to anything. From the products page to orders page, if these values are not passed in the URL, then this attack would not be possible.

## Fixing the Vulnerability

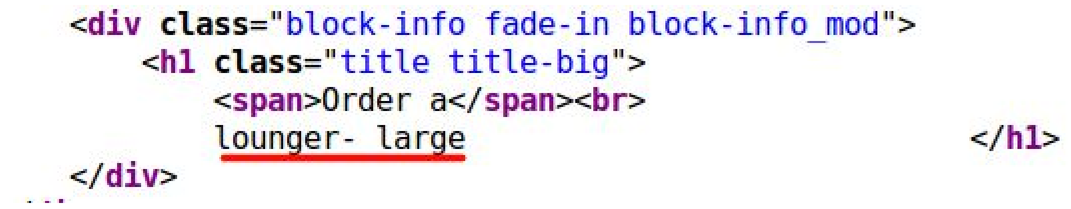


No: 109, Galle Road,
Dehiwala South
(Between Dehiwala Arpico & flyover onto the landside)
Parking Available

Call us or send us an email for quotations
E: info@beanbag.lk
M: +94 76 724 5288
T: +94 11 314 4288

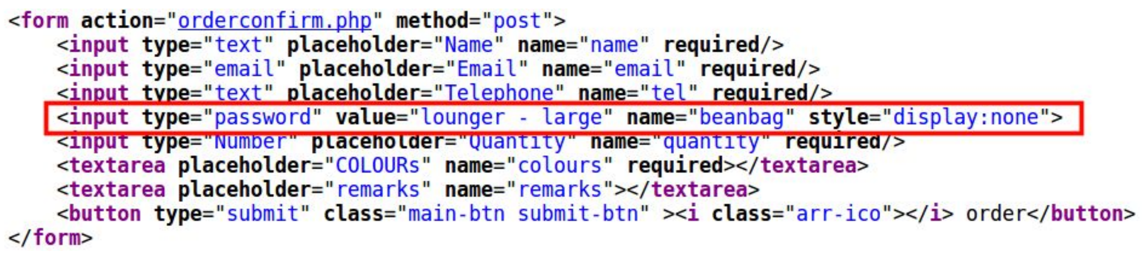
When viewing the page source of the order.php page, we can see that in two places, the query parameter values are used.

First place is the H1 element that displays the size and the type of the product.



```
<div class="block-info fade-in block-info_mod">
    <h1 class="title title-big">
        <span>Order a</span><br>
        lounger- large                              </h1>
</div>
```

Second place is a hidden password field where the value is set to the combination of the query parameter values. This hidden field may be used for internal purpose.



```
<form action="orderconfirm.php" method="post">
    <input type="text" placeholder="Name" name="name" required/>
    <input type="email" placeholder="Email" name="email" required/>
    <input type="text" placeholder="Telephone" name="tel" required/>
    <input type="password" value="lounger - large" name="beanbag" style="display:none">
    <input type="Number" placeholder="Quantity" name="quantity" required/>
    <textarea placeholder="COLOURs" name="colours" required></textarea>
    <textarea placeholder="remarks" name="remarks"></textarea>
    <button type="submit" class="main-btn submit-btn" ><i class="arr-ico"></i> order</button>
</form>
```

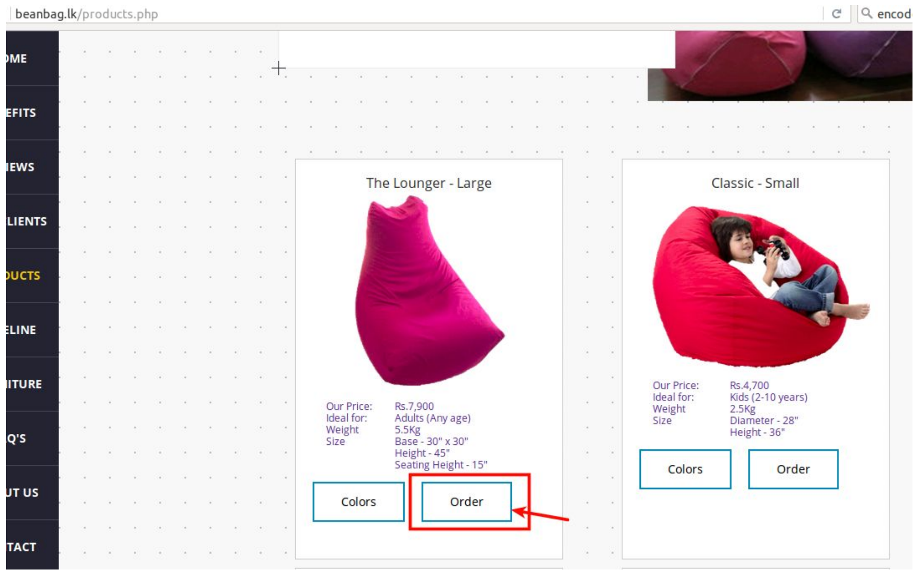
Normally, for preventing Cross Site Scripting vulnerabilities, the recommendation is to use proper encoding. You can use a library like OWASP_PHP_Filters for that. https://www.owasp.org/index.php/OWASP_PHP_Filters

The fix would be to read two query parameter values to variables and then sanitize the values using the appropriate method given in above library. Then finally display the sanitized content in the web page.

However, we cannot proceed with above approach in this matter. The reason is, if we follow the above approach, it would prevent Cross Site Scripting vulnerability, but still an attacker and inject some generic text like “We no longer sell Bean Bags” and get that displayed in the web page. ( http://beanbag.lk/order.php?size=Bean Bags&bb=We no longer sell )

Therefore, the correct fix that addresses both query string injection and cross site scripting is to avoid sending the two parameters in the URL. Which means, instead of a HTTP GET request, we need to send HTTP POST where the two parameters would be posted to the orders page from products page, but not included in the URL.

Currently in the products page http://beanbag.lk/products.php , the Order button calls the URL to orders page sending the query parameters http://beanbag.lk/order.php?size=large&bb=lounger .



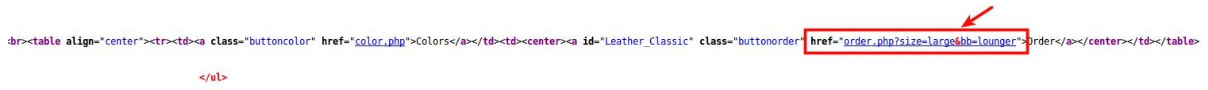
The Lounger - Large

| | |
|---|---|
| Our Price: | Rs.7,900 |
| Ideal for: | Adults (Any age) |
| Weight | 5.5Kg |
| Size | Base - 30" x 30"<br>Height - 45"<br>Seating Height - 15" |

Classic - Small

| | |
|---|---|
| Our Price: | Rs.4,700 |
| Ideal for: | Kids (2-10 years) |
| Weight | 2.5Kg |
| Size | Diameter - 28"<br>Height - 36" |

This can be seen in the page source too.



```
:br><table align="center"><tr><td><a class="buttoncolor" href="color.php">Colors</a></td><td><center><a id="Leather_Classic" class="buttonorder" href="order.php?size=large&bb=lounger">Order</a></center></td></table>
              </ul>
```

So, to correctly fix, for each item we can add a HTML form and set the action to order.php, and set the HTTP method to 'POST'. The two parameters 'size' and 'bb' can be added to the form as hidden fields and from order.php page, read the two parameters received in the POST request.

In addition to that, the fix can be further improved where proper sanitization (escaping) is done for the parameter values and then display on the web page.

## Declaration

I, Tharindu Edirisinghe hereby declare that all the security testing carried out on the beanbag.lk website were done only with the intension of discovering any potential security vulnerabilies of the website and to ensure the safety of the end users of the website. Apart from that, I hereby confirm that I had no intention of harming the website or the brand name of the company or gaining any financial benefit over exploiting the discovered vulnerabilities. This report is prepared for helping the web development team of BeanBag.LK and this will only be shared with the above stakeholders until the discovered issues are fixed.

Tharindu Edirisinghe,
Independent Security Researcher
Author of http://securityinternal.com
Twitter: https://twitter.com/thariyarox
LinkedIn: https://lk.linkedin.com/in/ediri