Beginner Guide to Insecure Direct Object References (IDOR)

hackingarticles.in/beginner-guide-insecure-direct-object-references

Raj July 4, 2017

Since 2013, the OWASP Top 10 Web application security risks list ranks Insecure Direct Object References (IDOR) fourth. This vulnerability allows an authorized user to obtain information from other users and can occur in any type of web application. Essentially, it enables requests for specific objects through pages or services without proper verification of the requester's rights to the content.

Definition by OWASP

Insecure Direct Object References allow attackers to bypass authorization and access resources directly by modifying the value of a parameter used to directly point to an object. Such resources can be database entries belonging to other users, files in the system, and more. This is caused by the fact that the application takes user-supplied input and uses it to retrieve an object without performing sufficient authorization checks.

The Application uses untested data in a SQL call that is accessing account information.

Let consider a scenario where a web application allows the login user to change his secret value.

Here you can see the secret value must be referring to some user account of the database.

Currently, user **bee** is login into a web server for changing his secret value. But he is willing to perform some mischievous action that will change the secret value for another user.

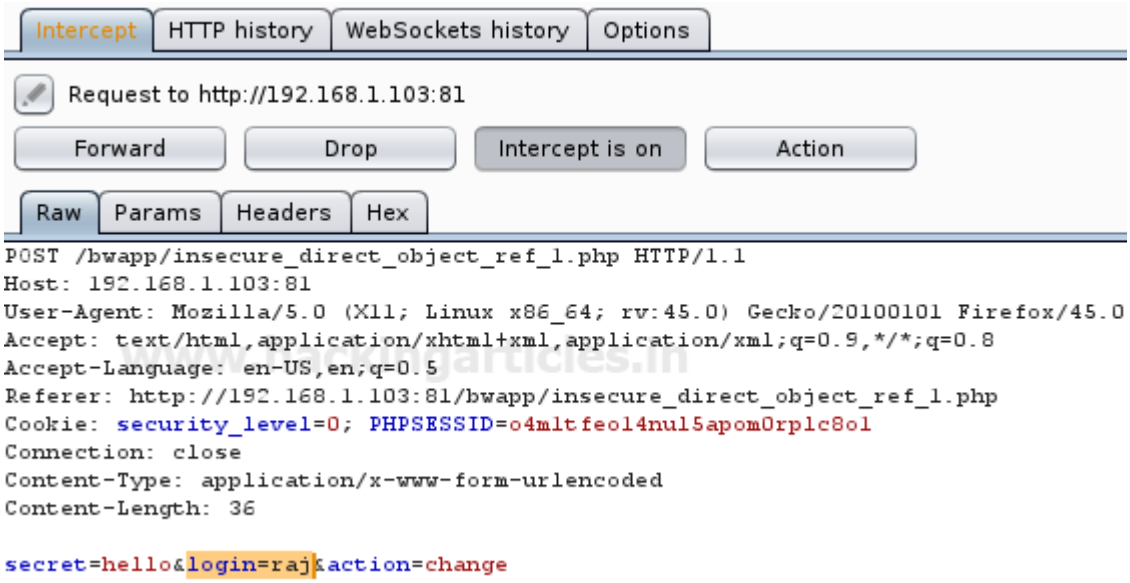
Using burp suite we had captured the request of the browser where you can see in the given image login user is the **bee** and secret value is hello. Now manipulate the user from another user.

SQLquery = "SELECT * FROM useraccounts WHERE account = ‘bee’;

```
POST /bwapp/insecure_direct_object_ref_1.php HTTP/1.1
Host: 192.168.1.103:81
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.1.103:81/bwapp/insecure_direct_object_ref_1.php
Cookie: security_level=0; PHPSESSID=o4mltfeol4nul5apom0rplc8ol
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 36

secret=hello&login=bee&action=change
```

Now let’s change user name into as shown in the given image. To perform this attack in an application it requires at least two user accounts.



Great!!!

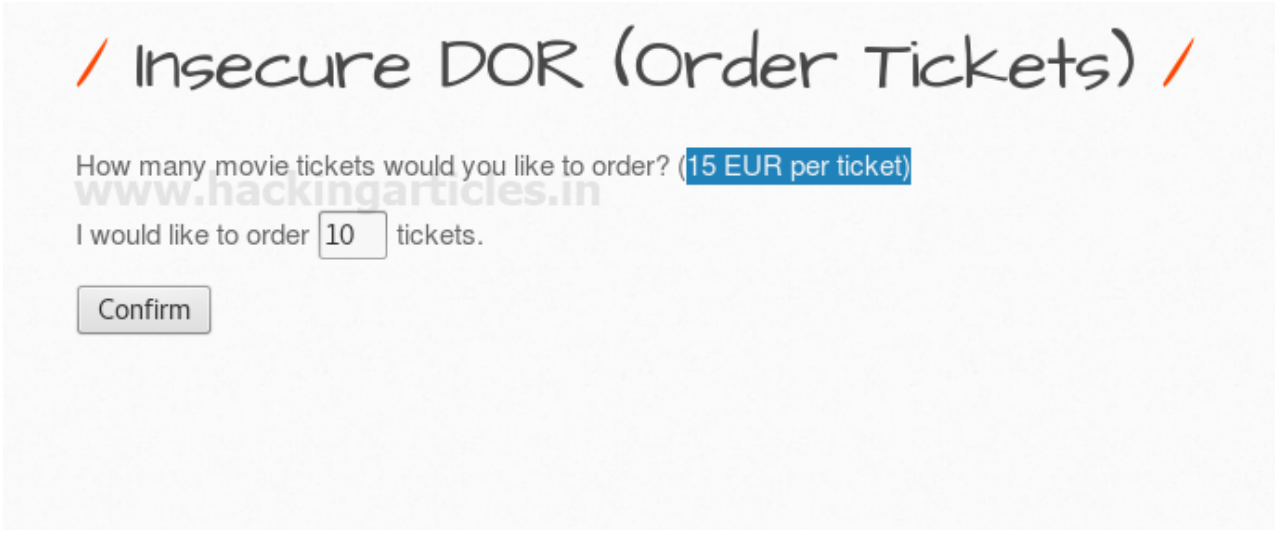
Note: in any official website the attacker will replace user account from an admin account.

Let take another scenario that looks quite familiar for most of the IDOR attack.

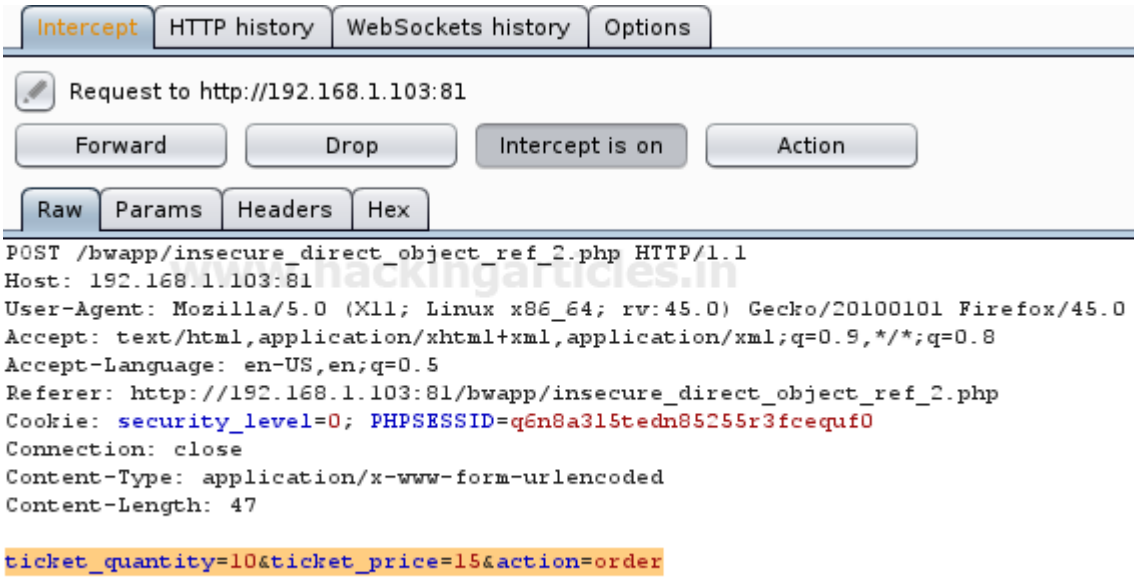
Many times we book different order online through their web application, for example, bookmyshow.com for movie ticket booking.

Let consider the same scenario in bwapp for movie ticket booking, where I had book 10 tickets of 15 EUR for each.

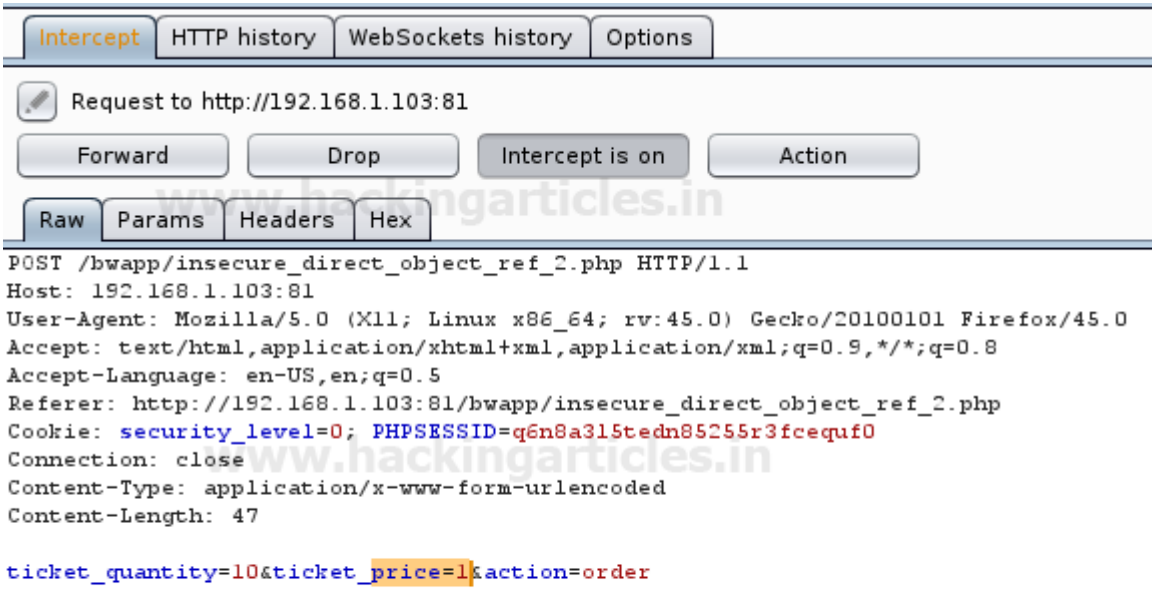
Now let's confirm it and capture the browser request through burp suite.



Now you can see we intercepted a request where the highlighted text contains a number of tickets and the price of one ticket, i.e., 15 EUR. It means it will reduce 150 EUR from my (user) account; now manipulate this price to your desired price.



I had changed it into **1 EUR** which means now it will reduce only 10 EUR from the account. You can observe it from a given image then forward the request.



Awesome!!! We had booked the 10 tickets in 10 EUR only.

To learn more about Website Hacking. Follow this **Link.**

Author: Aarti Singh is a Researcher and Technical Writer at Hacking Articles an Information Security Consultant Social Media Lover and Gadgets. Contact **here**