M183
Indirect Direct Object Reference

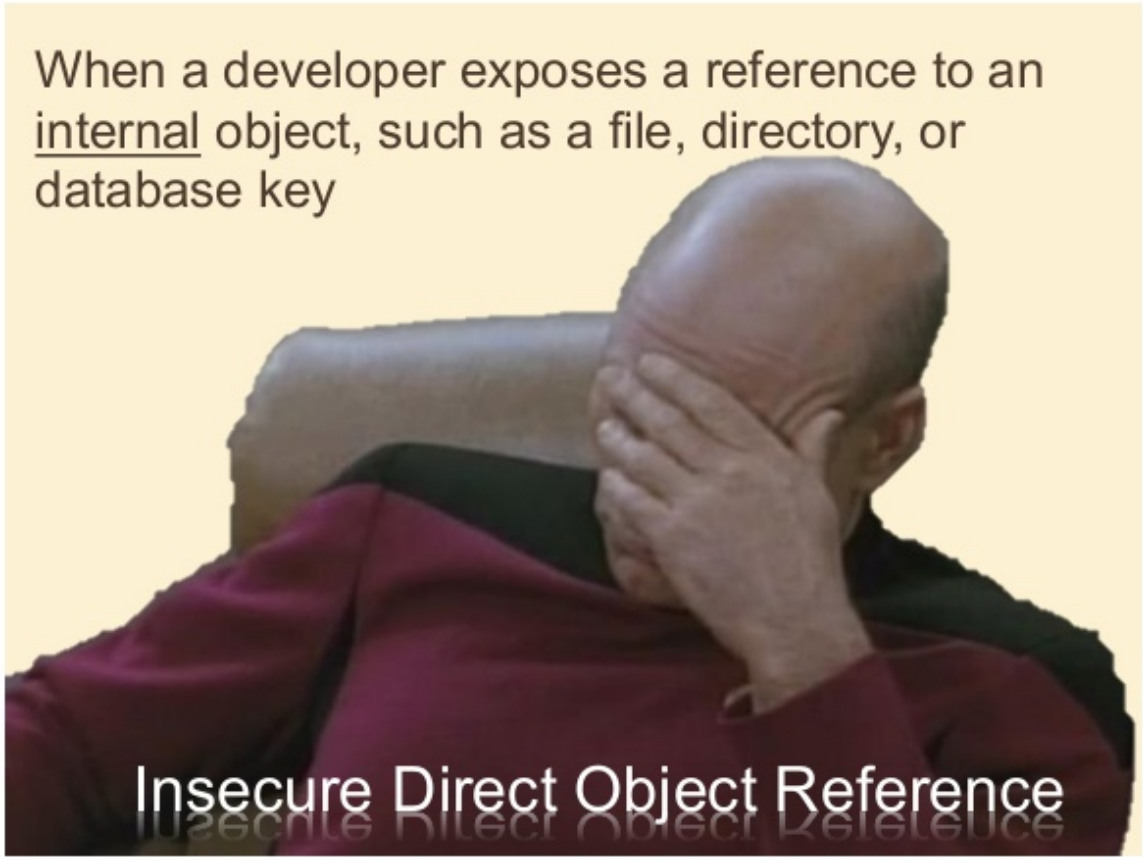
Timo Bonomelli, Patrick Günthard

February 18, 2016

Table of Contents

Table of Contents

What is *Insecure Direct Object References*



Threats

- **Threat Agents:** Any user who has only partial access to certain type of system data
- **Attacker's Approach:** Attacker, an authorized system user, simply changes a parameter value that directly refers to a system object to another object the user isn't authorized to use
- **Security Weakness:** Applications don't always verify the user is authorized for target objects

Example: Code

Example Website:

```
...

$conn = new mysqli(...);
$conn->query("UPDATE tbl_user SET password = '".$_GET['pw']."' WHERE username = '".$_GET['user']."';)


...
```

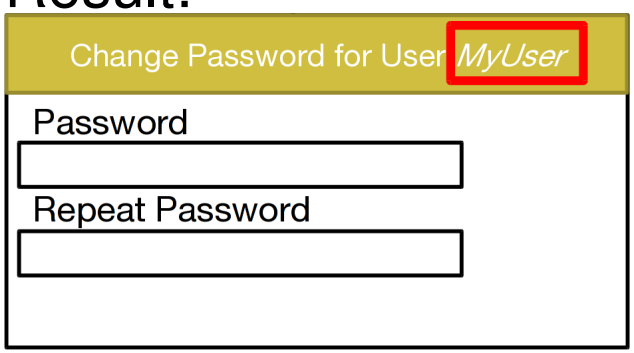
Example: Attack

Normal behavior

Example URL:

`http://somesite.net/changepassword?user=`*`myuser`*

Result:

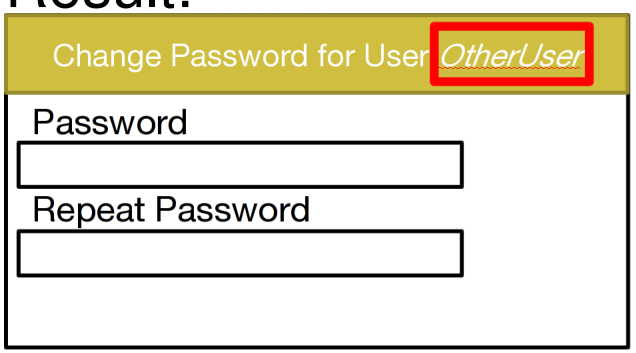


Attack behavior

Example URL:

`http://somesite.net/changepassword?user=`*`otheruser`*

Result:



Example: Attack

This URL:

`/changepassword?user=`*`otheruser'; DROP tbl_user;`* (1)

Table of Contents

Solutions and Problems

	Advantage	**Disadvantage**
Session Based	Only one authorization has to be done, access data for Database etc. is saved on the server and is not accessible by the attacker	A session uses a lot of memory for each user. For applications with a high number of users, a session for each client is not possible i.e. a non-session solution has to be implemented
Authorization	No Session is needed i.e. less memory is used and more users can access the application	Authorization is needed every time the user accesses data which is more complex to implement