

# Lightning Talk

## Indirect Direct Object Reference

Timo Bonomelli, Patrick Günthard

February 18, 2016

# Table of Contents

Vulnerability

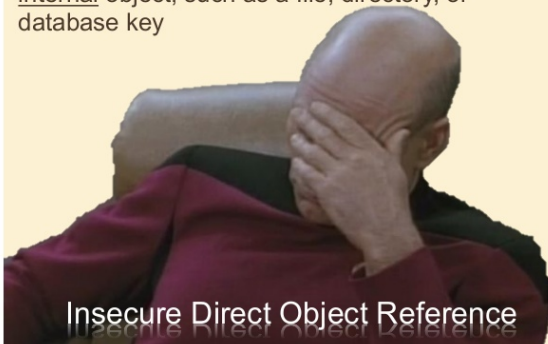
Threats

Example

How to prevent an Attack

# What is *Insecure Direct Object References*

When a developer exposes a reference to an internal object, such as a file, directory, or database key



Insecure Direct Object Reference

# 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/change  
password?user=myuser`

Result:

Change Password for User <i>MyUser</i>
Password
<input type="text"/>
Repeat Password
<input type="text"/>

### Attack behavior

Example URL:

`http://somesite.net/change  
password?user=otheruser`

Result:

Change Password for User <i>OtherUser</i>
Password
<input type="text"/>
Repeat Password
<input type="text"/>

## Example: Attack

This URL:

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

# Solutions and Problems

	<b>Advantage</b>	<b>Disadvantage</b>
<b>Session Based</b>	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
<b>Authorization</b>	TBD	TBD