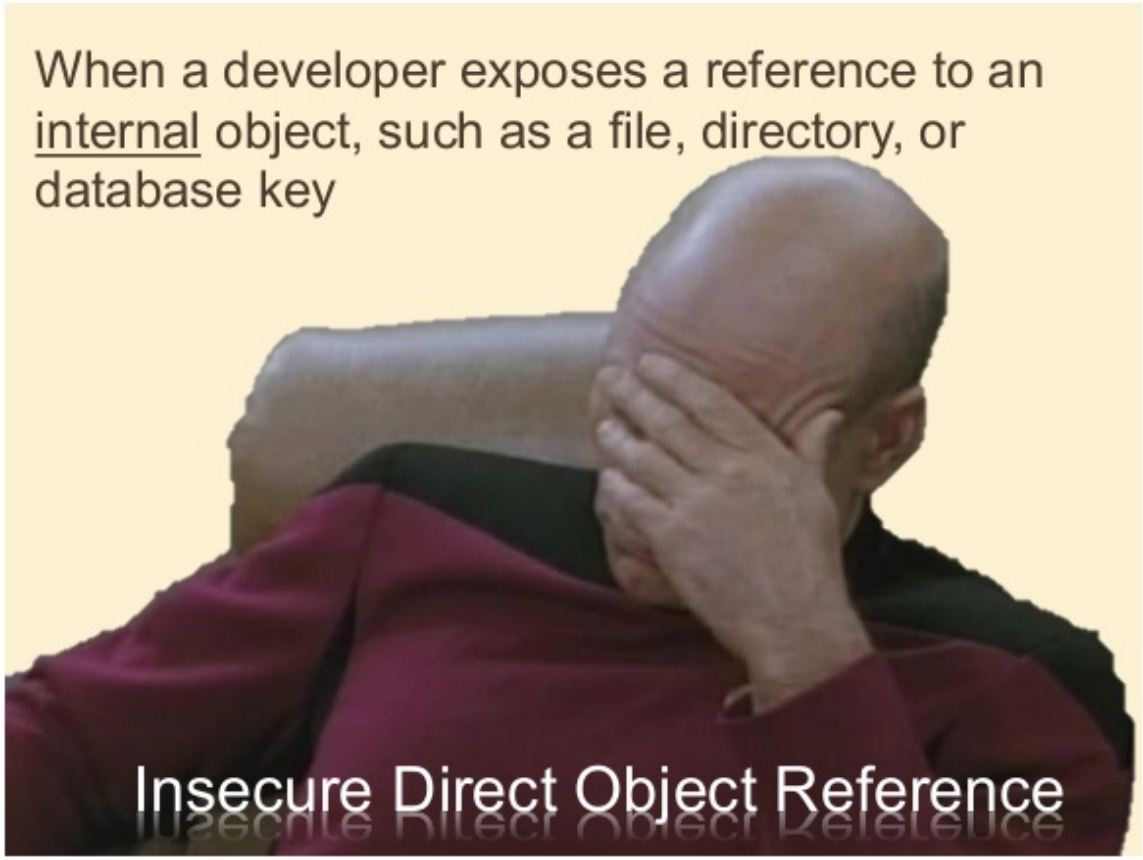
M183
Insecure Direct Object Reference

Timo Bonomelli, Patrick Günthard

February 19, 2016

Inhalt

What is *Insecure Direct Object References*



Gefahren

- **Von wem geht die Gefahr aus?** Jeder Benutzer welcher nur teilweisen Zugriff auf gewisse Daten hat
- **Angriffsansatz:** Der Angriffer, ein authorisierter System-User, ändert ein parameter welcher direkt mit einem Systemobjekt/Parameter referenziert zu einem anderen Objekt/Parameter, zu dessem Zugriff er nicht berechtigt ist.
- **Sicherheits-Lücke:** Applilationen überprüfen nicht bei jedem Zugriff die authorisierung

Example: Code

Beispiel Website:

...

```
String query = "SELECT * FROM accts WHERE account = ?";
PreparedStatement pstmt = connection.prepareStatement(query , ... );
pstmt.setString( 1, request.getParameter("acct"));
ResultSet results = pstmt.executeQuery();
```

...

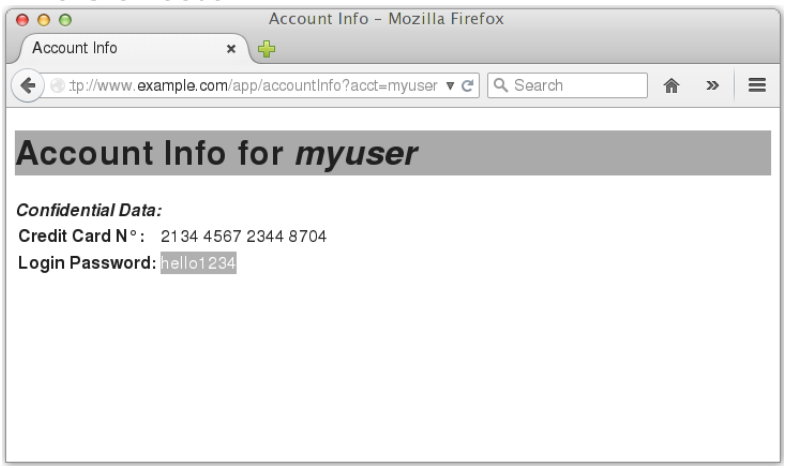
Beispiel: Angriff

Normaler Zugriff

Example URL:

`http://example.com/app/accountInfo?acct=myacct`

Resultat:

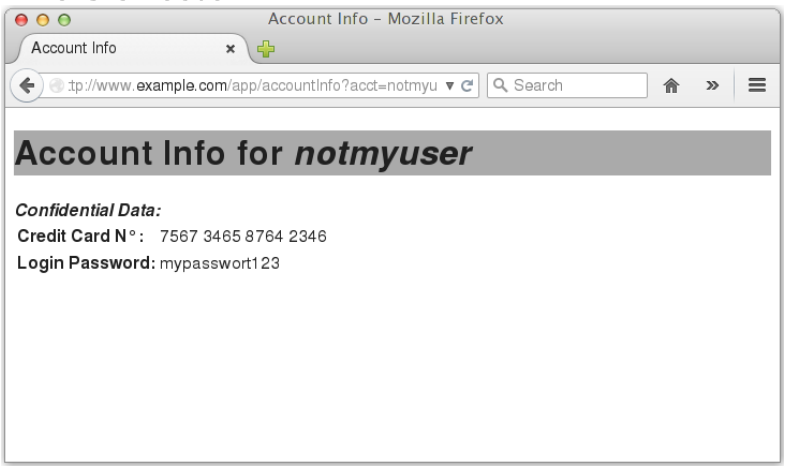


Angriff

Example URL:

`http://example.com/app/accountInfo?acct=notmyacct`

Resultat:



SQL Injection Attacke (OWASP A1)

URL:

`app/accountInfo?acct=`*`myacct'; DROP accts;`* (1)

Inhalt

Session

- Keine *Direct Object Reference* müssen an den client gesendet werden, sie können auf dem Server in der Session gespeichert werden.
- Wenn trotzdem Refernzen benötigt werden, können sie sich von den ursprünglichen daten unterscheiden und gemappt werden.

Authorization

- Every access is checked if the user is authorized to do that. Example: A random token can be created for each user which then is checked every time the user accesses the page

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

Inhalt

Summary

- Serious Issue
- Easily preventable
- Several fixing solutions

End

Sources:

- OWASP: 2010-A4-Insecure Direct Object References
- Wikipedia: HDIV