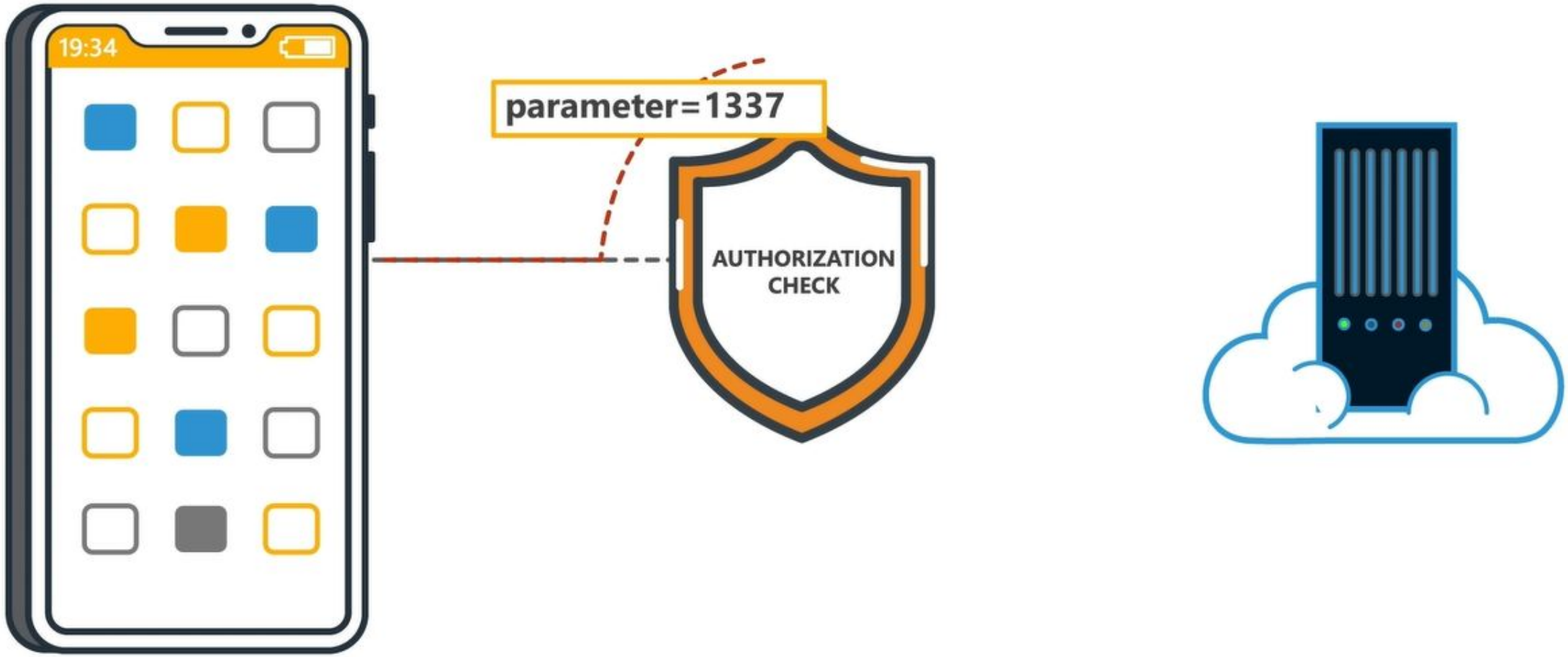
SECURE CODE WARRIOR

INSECURE DIRECT OBJECT REFERENCE
INSECURE AUTHORIZATION

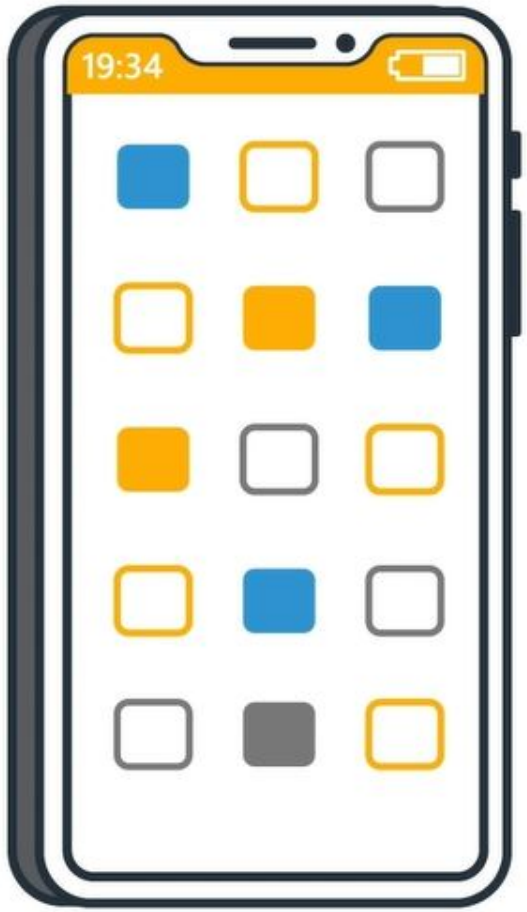
We'll also go through

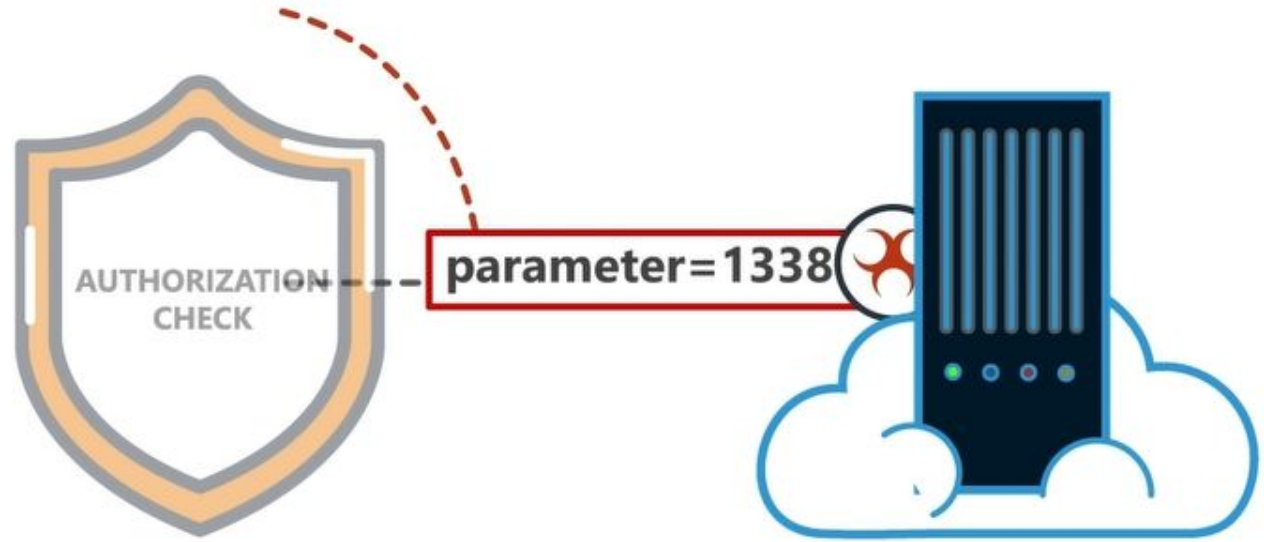
some of the causes and preventions
of this vulnerability

When an application gives users access to objects based on user-supplied input,



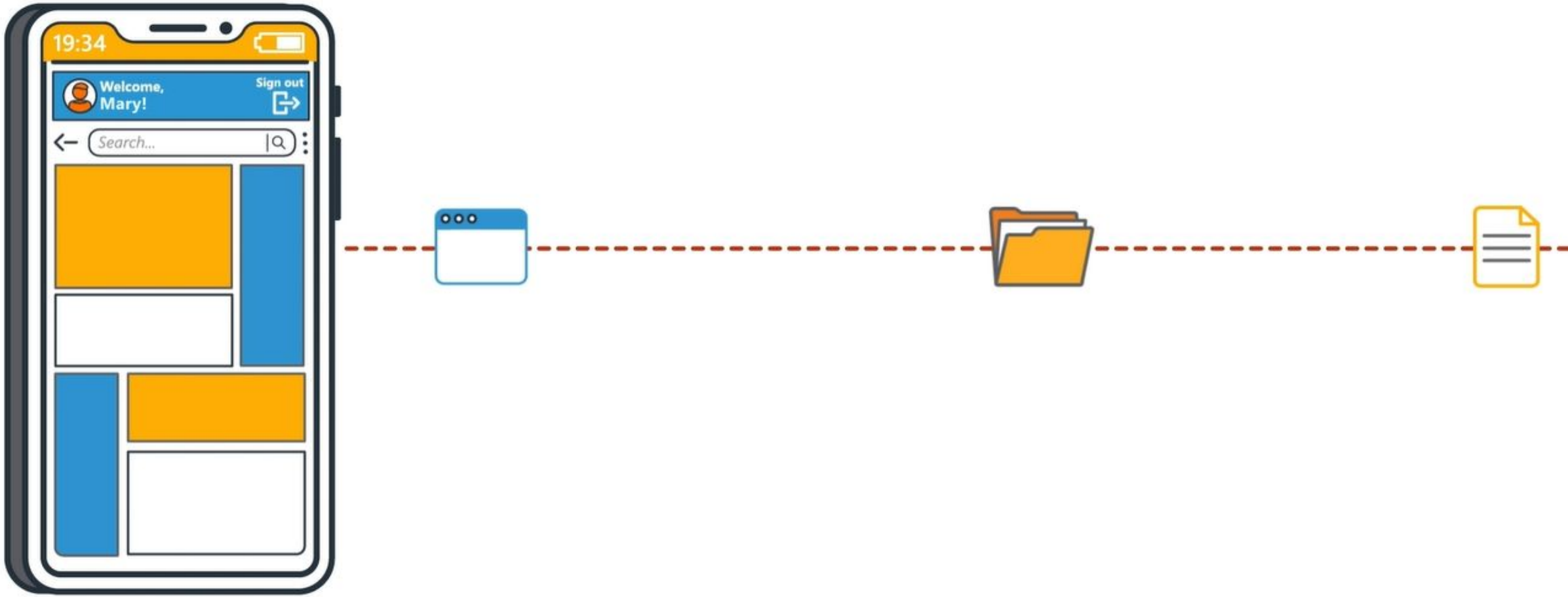
attackers can bypass authorization requirements and access resources directly in the system.





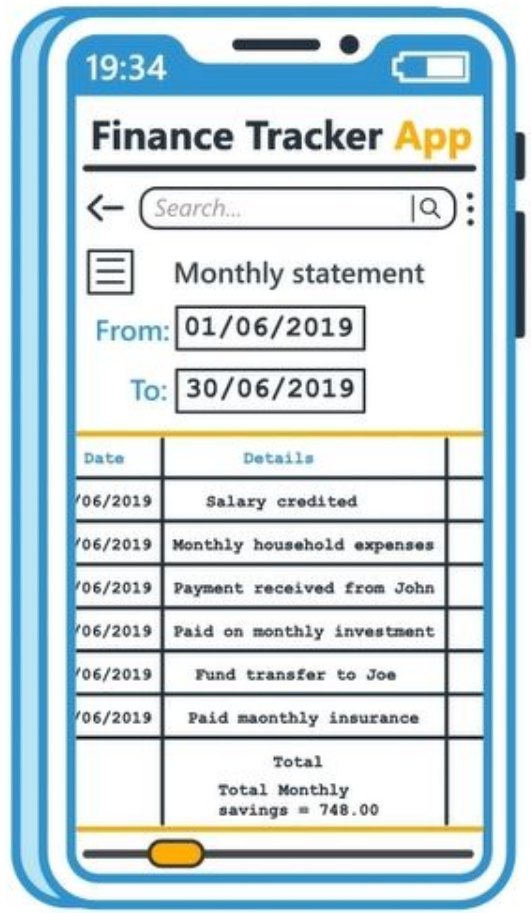
Generally this vulnerability is very easy to exploit. Attackers simply modify the value of an accessible parameter, such as the User ID or personal data, which directly points to an object.

In other words, a user could supply an input and thereby access, for example, other user profiles, user contact information, private messages, files, or other resources - without any check on authorization.



LET'S LOOK AT AN EXAMPLE

This personal finance app lets users create a budget and track expenditures and account data.

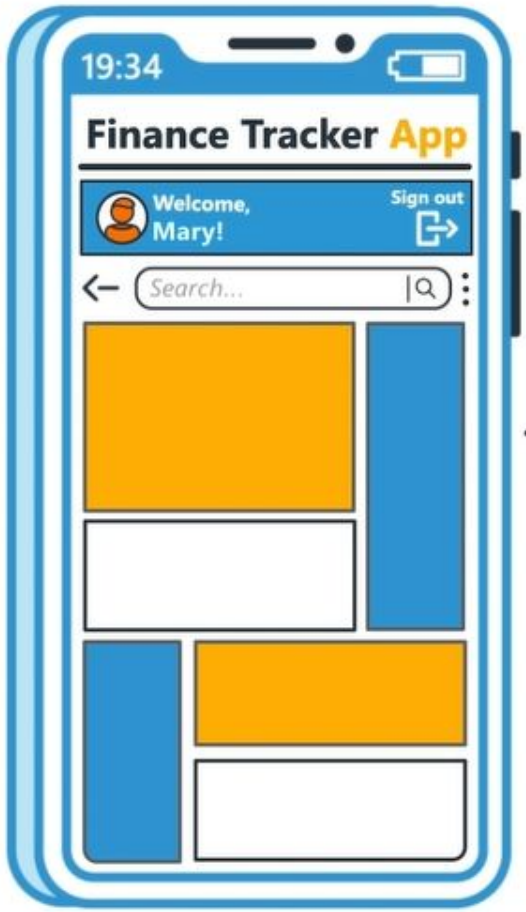


Unfortunately, it has no authorization checks on the backend API,

which has produced an insecure direct object reference.

https://app.financeapp.com/v1/companies?userid=736251993

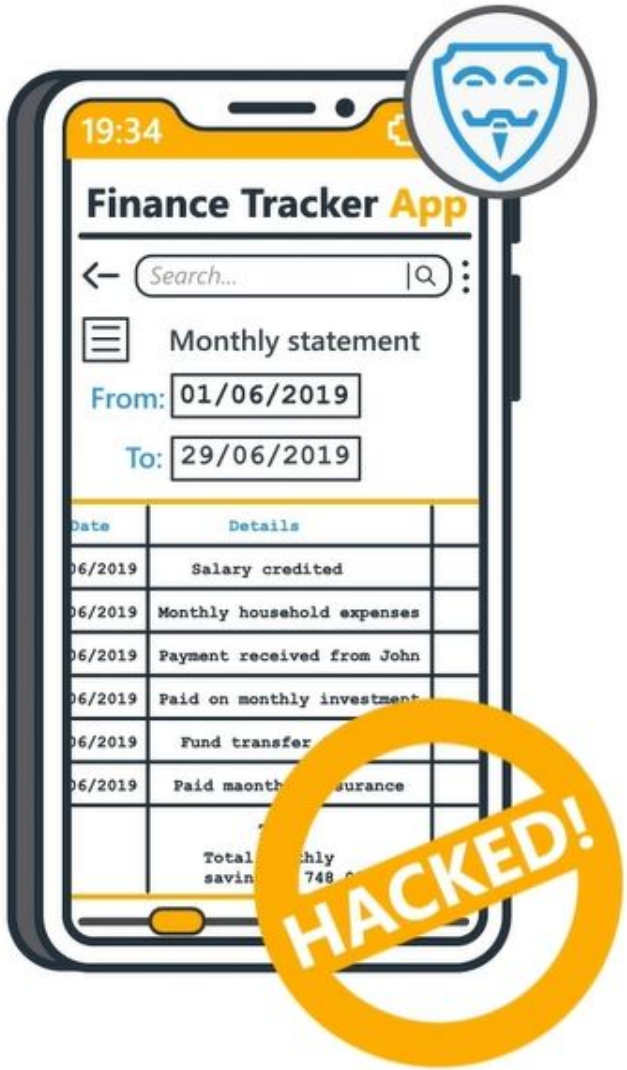
As a result, any logged in user is able to request and access other user profiles simply by manipulating the User ID parameter in the URL request.



https://app.financeapp.com/v1/companies?userid=736252993

An attacker takes advantage of this vulnerability to access the financial data of other users in the app.

Now, the confidential financial data of all users is exposed.



To prevent Insecure Direct Object Reference vulnerabilities, developers should:

- Ensure that the back-end API that the mobile device calls utilizes proper access control to the relevant end-points
- If applicable, test among users with different levels of authorization

Congratulations, you have now completed this module!

SECURE
CODE
WARRIOR