Access control is another term from authorization. This is basically how the application discerns weather a client should be able to access certain resources.

There are vertical and horizontal access controls, vertical being elevated privileges like admin type stuff, horizontal being able to access content that belongs to other users such as someone else’s shopping cart for example.

There is also context dependent access control:

Graphical user interface, text

Description automatically generated

Unprotected admin functionality with obscure URL:

Graphical user interface, text, application, email

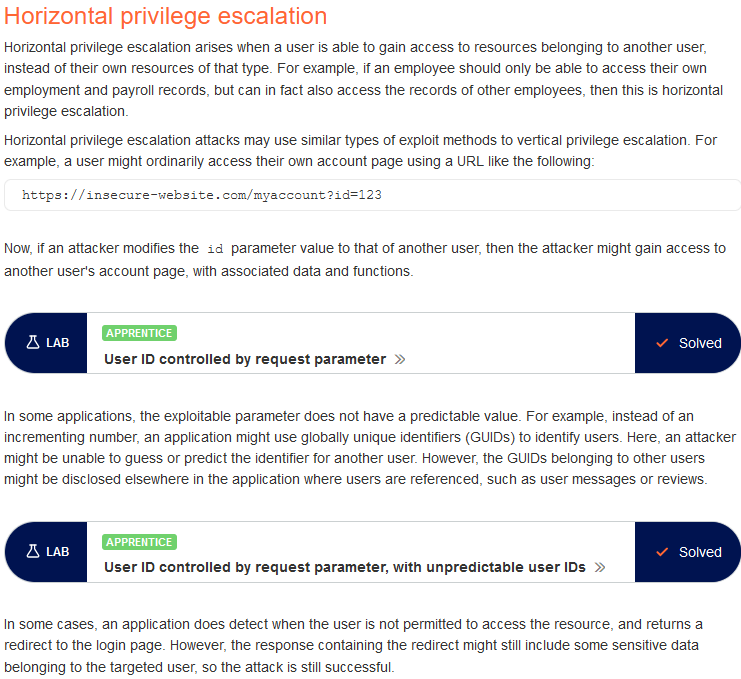
Description automatically generatedOf course always look out for sensitive comments, if theyre hiding their admin page behind an obscure URL if we find it were in.

Text

Description automatically generatedSimilar to above but this is just obscurity through parameters. Again think about trying to access other users as well besides just the admin or those with elevated privs. Accessing other accounts is still good!

Graphical user interface, text, application, email

Description automatically generatedChanging the HTTP method and or messing around wit the X-Original-URL header as shown can sometimes bypass access controls.



Test applications for this!!

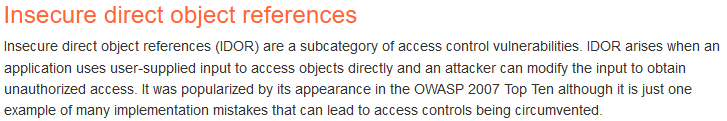
Use the container extension to color code different accounts and or open private windows, mess around and try to do some horizontal stuff.

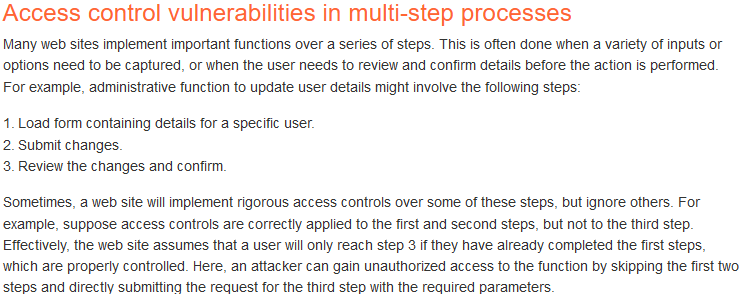
Try swapping CSRF tokens or other cookies …

OF course if we find horizontal try to escalate it into vertical

Text, timeline

Description automatically generated

SO basically the same thing here. User changes a URL parameter. Look for files being referenced or something and use the container extension and/or private windows for testing. Try to combine this with things like stored xss, by referencing a malicious file stored in the database by your other user account. A lot can be done.

If you can reach elevated functionality record the request in burp and then try to request it again with a cookie from an unprivileged or unauthenticated user.

Text

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

IDOR:

Basically a horizontal access control vulnerability whereby users of the same privilege level can access one anothers content. This can easily be tested for using the Autorize entension in burp, using one accounts cookies and logging into the other and browsing around. Autorize will automatically make the same requests were making just using the other accounts cookies.

This could also happen in the URL where we see a parameter for example id=1242 changing this ID could result in seeing another users information.