**Practical exercises and questions**:

In the following section there will be six exercises about Broken Access Control and Cross Site Request Forgery. We suggest you to follow the exercises in the order in which they are presented.

**Exercise 1**: In the following exercise you will experience what happens if administration privileges are managed by role in the application and this role can be easily changed by every user.
You can find the lab here: https://portswigger.net/web-security/access-control/lab-user-role-can-be-modified-in-user-profile

**Objective**: The goal of the exercise is to access the admin panel to delete the user named Carlos. Keep in mind that this lab has an admin panel ad /admin and this can be accessed only by users with a roleid of 2. Use these credentials to access the application: Username: wiener – Password: peter

**A suggestion in case you are lost**: *Look at the request and response parameters: you might be able to modify something…*

***N.B.**: To solve this lab you need to use some tools to intercept and modify HTTP requests. You can download Burp Suite Community Edition from here: https://portswigger.net/burp/communitydownload*
*You need to set Burp Suite as your proxy to intercept and manage requests.*

**Exercise 2**: In the following exercise you will experience what happens user identifiers can be easily modified and password easily found.
You can find the lab here: https://portswigger.net/web-security/access-control/lab-user-id-controlled-by-request-parameter-with-password-disclosure

**Objective**: The goal of the exercise is to access the admin account and delete the user named Carlos. Keep in mind that this lab has user account pages that contain the

current user's existing password prefilled in a masked input. Use these credentials to access the application: Username: wiener – Password: peter

**A suggestion in case you are lost**: *Try to view the administrator's account and find the password. It might be easier than you think!*

**Exercise 3**: In the following exercise you will experience what happens if access control can be easily circumvented by modifying some parameters.
You can find the lab here: https://portswigger.net/web-security/access-control/lab-url-based-access-control-can-be-circumvented

**Objective**: The goal of the exercise is to access the admin account and delete the user named Carlos. Keep in mind that this lab has an unauthenticated admin panel at /admin, but the frontend system has been configured to block external access to that path. However the backend application is built on a framework that supports the X-Original-URL header.

**A suggestion in case you are lost**: *To delete a user you need to make a request to the right URL providing the right username. You can modify request parameters and headers to suit your needs…*

**Exercise 4**: In the following exercise you will experience what happens if validation parameters are used only in certain types of requests.
You can find the lab here: https://portswigger.net/web-security/csrf/bypassing-token-validation/lab-token-validation-depends-on-request-method

**Objective**: The goal of the exercise is to change the viewer's email address using some HTML that performs a CSRF attack. The only thing you know is that the email change functionality is vulnerable to CSRF. It attempts to block CSRF attacks but the protection is applied to certain types of requests. Use these credentials to access the application: Username: wiener – Password: peter

**A suggestion in case you are lost**: *Try to understand how a legitimate request works. Maybe a different request method can work too… You can write a pre-compiled request to be automatically sent when the victim visits your page. Use the exploit server to deliver the attack.*

**Exercise 5**: In the following exercise you will experience what happens if tokens are not used wisely.
You can find the lab here: https://portswigger.net/web-security/csrf/bypassing-token-validation/lab-token-validation-depends-on-token-being-present

**Objective**: The goal of the exercise is to change the viewer's email address using some HTML that performs a CSRF attack. The only thing you know is that the email change functionality is vulnerable to CSRF. Use these credentials to access the application: Username: wiener – Password: peter

**A suggestion in case you are lost**: *Try to understand how a legitimate request works. Maybe not all the parameters are necessary to complete the request… You can write a pre-compiled request to be automatically sent when the victim visits your page. Use the exploit server to deliver the attack.*

**Exercise 6**: In the following exercise you will experience what happens if tokens are used correctly but are not bonded to users.
You can find the lab here: https://portswigger.net/web-security/csrf/bypassing-token-validation/lab-token-not-tied-to-user-session

**Objective**: The goal of the exercise is to change the viewer's email address using some HTML that performs a CSRF attack. The only thing you know is that the email change functionality is vulnerable to CSRF. It uses tokens to try to prevent CSRF attacks but they are not integrated into the site's session handling system. Use these

credentials to access the application: Username: wiener – Password: peter and Username: carlos – Password: montoya

**A suggestion in case you are lost**: *Try to understand how a legitimate request works. Do all the parameters have to be unique? Do they have to be tied to the user's session? If you want to know something useful try to look at the page source, you may find something interesting…*
*Use the exploit server to deliver the attack.*

**Questions**:

In this little section you will find some simple questions to test your comprehension of the topic.

1. Name some Broken Access Control vulnerabilities and briefly describe them.
2. Simply describe the necessary steps for a CSRF attack.
3. Explain how a malicious request can be embedded in a web page and be automatically sent when visiting it.