All this text wasn’t written by me (Tiago a.k.a TigaxMT).

All of it was transcribed by me from HackerOne video lessons.

All credits and thanks go to them.

Cookie Tampering

MANIPULATING THE BROWSER

The easiest way to manipulate cookies on the browser side is via Firefox’s development tools. The ‘cookie’ command is extensive and allows you to list, delete, edit and add cookies arbitrarily.
This makes it trivial to exploit the likes of level2 where you’re changing the admin cookie.

MANIPULATING RESPONSES

My standard technique is to manipulate the response coming from the server, using Burp, to set cookies as need be. In this, you either edit Set-Cookie headers, or add new ones as you see fit.
Word of warning: It’s very easy to miss the first cookie assignment, and run into issues trying overwrite it. This technique works best if you’re intercepting all the requests in a given time period.

FLAGS

Whenever you’re tampering with cookies, make absolutely sure that you check the HTTPOnly and Secure flags. This is often a source of good low-hanging-fruit in an application, as many do not adequately set this.

The HTTOnly flag can also make it more difficult for you tamper with cookies on the client side, though of course that will only apply from JS.

DATA ENCODING

When looking at cookies, I recommend you always decode the data that is present, if it’s encoded in any way. Tips:

- If it ends in = or contains / it’s almost definitely Base64’d
    - Occasionally you’ll see _ instead of / and – instead of +, to get around URI encoding
- If it is all range 0-9A-F and all in either upper or lower case, it’s almost definitely hex encoded.

MANIPULATIONS

Common ways to manipulate cookies:

- Change individual bits of data to see the results

- Swap key-value pairs, or add duplicates; for instance, if you have a single cookie containing foo=bar&baz=test, try swapping the order of these pairs , or add an additional &foo=something to the end
    - Often, the first will pass some initial validation, and the second will be used for another purpose.

HASHES

If you find yourself with a cookie ending in 32-40 nibbles of hex, there’s a decent chance this is a hash. If it’s done properly (see HMAC in crypto lessons) then this should be safe; you can’t tamper with the data.
However, if it’s a bare hash, it may be possible to perform a length extension attack. This would let you append arbitrary data to the end of the cookie.

NO RULES

There are no hard and fast rules here. Just modify the data. What happens if a hex encoded cookie isn’t sent in hex format? Might be able to trigger an error on the server, giving you valuable data.
What if you edit the name of fields? Does it matter at all? This can give you insight into how the code parsing the cookie works.