* WRT Cookies

They are generally URL-encoded. But it is not mandatory. RFC specifies that whitespace, comma and semicolon must be encoded.

They are case sensitive.

* When uploading a file

When you are writing client-side code, all you need to know is **use multipart/form-data when your form includes any <input type="file"> elements**. multipart/form-data is significantly more complicated but it allows entire files to be included in the data. File is sent as part of body of HTTP Request.

* In case of redirects, use *noredirect* plugin of firefox to prevent HTTP redirect (Status Code 3XX).
* Prima Facie, HTTP headers are not url encoded.
* If a webserver is supporting CGI-BIN, check if it vulnerable to shell-shock.
* In case there is tomcat running,
  1. check if tomcat-manager page {*http://<ip-address>:8080/manager/html* } is accessible.
  2. Use Tomcat-users.xml file to get login details for various tomcat users.
* LFI

Apart from passing file name in the url, you execute commands locally using HTTP Header *User-Agent.* Refer [to](https://resources.infosecinstitute.com/local-file-inclusion-code-execution/) . ShellShock vulnerability is good example.

* *robots.txt* is a great place to look for vulnerability.
* Remember, text box is not the only way to provide user input. We can do so using drop-downs, checkbox, etc. Hence, these are generally overlooked for input sanitization. You can exploit these input-sources using proper tools.