ATTACKS:

1) CROSS SITE SCRIPTING:

Here client when logs into the server with his username and password. Then webserver sends the cookie back to the client which contains info about the username, password etc about the client. Then in the third step client sends back the cookie information back to the webserver and only this webserver can read this cookie as this cookie was initially generated by this server only. So this is called same origin policy that is cookie can be accessed only by the same server that generates it.

But java scripting can be used to get the user and logging information about the client.

For example on the amazon site if the bad guy has put up a some review about the some product and in that review he has also put up the java script code. And whenever any client reads that review containing the java script from the server. So that java script gets installed on the client in the form of a cookie. So that java script has the capability to read the information about the username and password of the client from the cookie and post that on the webserver from where the attacker can know the username and password of the client. This attack is shown in the diagram below:
It is another form of the command injection attack. It contains both the data and the code. Data is cookie and code is java script.

DEFENSES TO STOP THIS ATTACK:

1) FILTER: We can also use a filter which can look for <, > these brackets as java scripts normally is written in these brackets but this is not a good defenses as attacker might use a different strategies as attacker might use %3c which is also gets converted into these brackets <,>. But filtering method can be used either by black list or white listing. But in this case we must use white listing strategy.

2) When the webserver sends the cookie to the client then it can encode the ip address of the client in the cookie so even if bad guy gets the cookie he can’t match the ip address in the cookie

CROSS SITE REQUEST FORGERY:
So cross site forgery attack is an attack in which users are fooled to open link on the forged website while they are still in session with some authenticated webserver. For example you are tranfering money online from your bank. Then after logging in cookie are generated and when you click on the say transfer amount button then cookies are also sent along with it.

But if bad guy makes a similar website of that of the bank and fools you to click on link on that site created by bad guy and if you are still in a session with the authenticated website then cookie which was being generated when you communicated with the authenticated bank site may get send along when you click on that site send by the bad guy.

3) ERROR MESSAGE ATTACK:

Whenever we submit a invalid query we get a response back from the server as an error message.

For ex: if the attacker sends a query like SELECT * from user where name =" "

Then if the server returns the error message back to the user with enough error information then attacker may figure enough information out from the error messages about the data base maintained by the server and hence can carry out his attack.

4) TRACTOR BEAMING ATTACK: Process have real user id and effective user id.

For example real user id is set to 1001

    Effective user id be 1001

ftp server: after we login the server
real user id becomes 0
effective let say becomes 1001
SITE EXEC (location in FTP server where u can run commands)
Lets consider the following example.

Do special cmd(....){
    Set uid(0)
    If(..)
    Throw exception
    Set uid(1001)}
Event loop()
{ try {
    Do special command()
Now suppose in above program do special cmd will get called and uid will set to 0 but uid and then exception is called so uid will never set to zero again.

DEFENSES AGAINST THIS ATTACK.

1) add a command that set uid=1001

2) transaction : So if any exception occurs then OS will undo all the steps that it has done till now setting the uid back to its original value.