Discussion on some of the errors from CWE/SANS TOP 25 Most Dangerous Software Errors

#1. Cross Site Scripting (XSS)

**Attack:**

Client 1. username, pwd  
2. Cookie  
Attempts to contact the user  
Bad Guy  

Client 4. runs JS on user's machine  
5. The JS writes Cookie back as another review  
3. Writes a Review  
Server (WWW)  
6. Gets user's a/c information  
Bad Guy

Bad Guy (BG) attempts to contact the user because fails to do so because he doesn’t have a cookie to validate his session.

Thus, to connect to the user, BG writes on the Server (WWW) page, lets say a product review at Amazon.com. What he actually writes on the server’s page is a javascript which runs on the user's machine as soon as the user accesses that page. This js has a code to write back a review on amazon's page from the user. This review contains the user’s cookie for conxn to amazon allowing the BG to gain access to amazon using your account.

**Solutions:**

- Filter specific tags like `<a href>` (Can be bypassed)
- Use custom tags
- Prefer using whitelists over blacklists because whitelists are more exhaustive.
#2. Cross Site Request Forgery

**Attack:**

<table>
<thead>
<tr>
<th>Client</th>
<th>1. username, pwd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2. Cookie</td>
</tr>
<tr>
<td></td>
<td>4. Transfer request sent as a user's request by BG through user's account.</td>
</tr>
<tr>
<td>Bad Guy</td>
<td>3. Link fed with values to direct the bank to transfer money to BG's a/c</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Server (WWW)</td>
<td></td>
</tr>
</tbody>
</table>

Bad Guy (BG) tries to make the user click a link/button that contains an already cooked link that initiates a transfer request on user's behalf (if the user is connected to the bank at the time of attack). If BG is lucky, the user clicks the link when he has logged into his bank a/c. The link click initiates a transfer request from the user's a/c to the BG's a/c.

The BG never really contacts the bank directly but in turn uses the user to perform the attack. The bank is unable to detect an attack as it is only dealing with the user who is a trusted connection.

**Solutions:**
- Ask the user to confirm the transaction before the actual money transfer occurs. (Burden on user)
- Every interaction b/w the bank and the user should follow handshaking mechanism (credential verification).
- Use large random numbers in the URLs to avoid easy decoding by BG.

#3. Error Messages

**Attack:**

```
ERROR:
Msg: Unable to search for requested column on table USERS!
*(Followed by description of table/some columns.
```

Bad Guy (BG) tries to a hoax search on your database which results in an error. Sometimes the Errors are so detailed (generally used to help programmer find the bug easily) that the BG can attack your system using the details mentioned in a few errors. Commonly exploited by the encryption pattern of several weak encryption schemes.

**Solutions:**
- Avoid detailed description of Errors. Use standard Error statements.
- Avoid writing your own encryption algos. Use existing reliable encryption schemes.
#4. Exceptional Conditions

Tractor beaming Attack

Attack:

In UNIX, processes have 2 user IDs:
- Real UID (ruid)
- Effective UID (euid)

FTP Server
- After Login
  - ruid 0
  - euid

SITEEXEC (location in FTP Server where you can run commands)

dospecialcmd(…)
{
  set ruid(0)
  if(…)
  {
    throw exception
    seteuid(1001)
  }

  eventloop()
  {
    try
    {
      dospecialcmd()
      catch(e)
      {
      }
    }
  }
}