CS5331 Web Security Midterm Practice

Question 1 (Short answer questions)

a) If an attacker is able to control the response to users' DNS requests, list two threats faced by web users from this attacker?

1) web users could be tricked into giving their personal login information/credentials to an attacker controlled website thinking that is was the original website that they were searching for (stealing credentials)
2) attackers can act as a man in the middle of the web user and the legitimate server, this can threaten the web user's privacy

b) Explain how users can make sure that they are visiting the authentic website using certificate of web sites.

users can firstly check their browser for a green lock indicator at the top of the browser where the link of the website is. The user's browser can then check against a Certificate Revocation List or Online Certificate Status Protocol to see if the CA that signed the certificate has been compromised and if the certificate has been revoked.

c) Describe how two websites, *a.com* and *b.com*, use third-party cookies from *c.com* to track users.

They can both allow c.com to have an image on their websites. When the user's browser loads the sites from a.com and b.com, they will also request the image from c.com which can come attached with a unique cookie from c.com. This for a.com and b.com can receive the user's unique cookie and can collude to piece together information about the user.

Question 2 (Multi-choice Questions)

Under the same origin policy, decide whether JavaScript from the following sites can send request to `https://gmail.com` (By default setting, HTTP runs over TCP port 80, while HTTPS runs over TCP port 443). Circle Yes or No.

(Yes / No) `http://gmail.com`
(Yes / No) `https://mail.google.com`
(Yes / No) `https://gmail.com:443`
(Yes / No) `https://sg.gmail.com`

Question 3.

Given the following PHP code segments in an web application, which takes the parameter *group* from the URL:

```
<p>Hello user, your current group is beginner [ <?php echo
$_GET['group']; ?> ] </p>
```

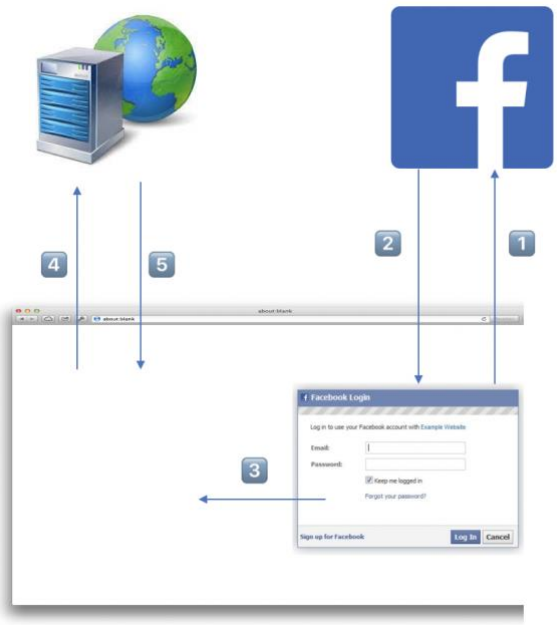
Please describe the vulnerability in this code. Describe how an exploit works and give two ideas of solutions to prevent it.

This code allows for a Reflected XSS attack. An attacker can trick the user to clicking on a link that will fill in the group parameter with a script which embeds a resource from the attacker and then send a cookie from the webpage back to the attacker controlled resource.
1st solution) input sanitisation or encoding to prevent special characters or terms from being sent immediately from the URL into the php function.
2nd solution) Whitelist to only allow certain words to be considered a 'group'

Question 4

A web site *A* uses Facebook's Web SSO service to authenticate its users. The following steps happen during the login process.

1) The Facebook iFrame sends users's username and password to the Facebook server.
2) After authentication, the Facebook server sends back the authentication token *T* and the user's public id *U*.
3) The Facebook iFrame sends *T* and *U* to the website through postMessage().
4) The web page from *A* displays information about *U* in the browser, and sends *T* and *U* to its server.
5) A's web server verifies the validity of *T*, and set a cookie to remember the authenticated session.



In Step 3 and 4, identify the possible attacks when using postMessage() to perform inter-domain communications.

If the origin of post message is not properly defined, another website can hold website A in an iframe, post a message to website A in the iframe and later receive T and U back from the website from step 3. Thus the attacker just needs to trick a user to use OAuth into website A from the attacker's own webpage.

Question 5

a) One of the key reasons for Cross-Site Request Forgery (CSRF) attacks is that the internal interface of a web application is publicly reachable. Therefore, extra checks before invoking important internal interfaces can help to prevent CSRF in web applications. Describe a solution that prevents CSRF attacks using this principle.

Validate the HTTP Origin of the referrer. This way, only allow access to the internal interface of a web app if the request was sent from a trusted HTTP referrer, which could be from its own domain.

b) This type of attack is not limited to the web browser platform. Please identify another similar attack on other environments, such as Android. Explain why they are similar even though the environments are different.

Andriod apps can refer to each other through a link, for eg: clicking a link online can open an app like joining a whatsapp group. Thus an attacker can trick a user to click a link to open the app on their phone with the user's session already logged in.