Cross Site Request Forgery

W2.1. Explain why the same-site cookie can help prevent CSRF attacks.

W2.2. Explain how a website can use secret token to prevent CSRF attacks, and why does it work?

W2.3. These days, most of the websites use HTTPS, instead of HTTP. Do we still need to worry about CSRF attacks?

W2.4. Using LiveHTTPHeader, we find out that the following GET request is used to send an HTTP request to `www.example.com` to delete a page owned by a user (only the owner of a page can delete the page).

```
http://www.example.com/delete.php?pageid=5

GET /delete.php?pageid=5
Host: www.example.com
...
```

Please construct a simple malicious web page, so when a victim visits this web page, a forged request will be launched against `www.example.com` to delete a page belonging to the user.

W2.5. Using LiveHTTPHeader, we find out that the following POST request is used to send an HTTP request to `www.example.com` to delete a page owned by a user (only the owner of a page can delete the page).

```
http://www.example.com/delete.php

POST /delete.php HTTP/1.1
Host: www.example.com
...
Content-Length: 8
pageid=5
```

Please construct a simple malicious web page, so when a victim visits this web page, a forged request will be launched against `www.example.com` to delete a page belonging to the user.

W2.6. The forged HTTP request against Elgg in this chapter needs Boby's user id (`guid`) to work properly. If Alice targets Boby specifically, before the attack, she needs to find ways to get Boby's user id. Alice does not know Boby's `Elgg` password, so she cannot log into Boby's account to get the information. Please describe how Alice can find out Boby's user id.

W2.7. In a request, there is an user id, which is a random number generated by the server. The ID information can be found from the user's page from the server. If an attacker does not know this user ID, can he/she still launch an CSRF attack on this service?

W2.8. If Alice would like to launch the attack on anybody who visits her malicious web page. In this case, she does not know who is visiting the web page before hand. (1) Can she still launch a CSRF attack to modify the victim's `Elgg` profile? Please explain. (2) Can she launch a CSRF attack to add her to the victim's friend list? Please explain.

W2.9. When a web page sends a request to its server, the session ID is always attached in the cookie section of the HTTP header. A web application requires all the requests from its own page to also attach the session ID in its data part (for GET requests, the session ID is attached in the URL, while for POST requests, the session ID is included in the payload). This sounds redundant, because the session ID is already included in the request. However, by checking whether a request has the session ID in its data part, the web server can tell whether a request is a cross-site request or not. Please explain why.

W2.10. Do browsers know whether an HTTP request is cross-site or not?

W2.11. Do servers know whether an HTTP request is cross-site or not?

W2.12. Why cannot a web server use the referer header to tell whether a request is cross-site or not?

W2.13. Why is it important for a server to know whether a request is cross-site or not?

W2.14. Can we simply ask browsers not to attach any cookie for cross-site requests?

W2.15. ★★★
If a page from `www.example.com` contains an `iframe`, inside which a facebook page is displayed. If a request is sent from inside the `iframe`, is it considered as a cross-site request or not? If not, how can be this secured?