Question 1 ***Boogle*** ()

Boogle is a social networking website that's looking into expanding into other domains. Namely, they recently started a map service to try their hand at fusing that with social media. The URL for the main website is `https://www.boogle.com`, and they want to host the map service at `https://maps.boogle.com`.

Q1.1 For each of the following webpages, determine whether the webpage has the same origin as `http://boogle.com/index.html`, and provide a brief justification.

i. `https://boogle.com/index.html`

ii. `http://maps.boogle.com`

iii. `http://boogle.com/home.html`

iv. `http://maps.boogle.com:8080`

Q1.2 Describe how to make a cookie that will be sent to only Boogle's map website and its subdomains.

Q1.3 How can Boogle ensure that cookies are only transmitted encrypted so eavesdroppers on the network can't trivially learn the contents of the cookies?

Q1.4 Boogle wants to be able to host websites for users on their servers. They decide to host each user's website at `https://[username].boogle.com`. Why might this not be a good idea?

Q1.5 Propose an alternate scheme so that Boogle can still host other users websites with less risk, and explain why this scheme is better.

Note: It is okay if the user sites interfere with each other, as long as they cannot affect official Boogle websites.

Question 2 ***Cross-Site Request Forgery (CSRF)*** **()**

In a CSRF attack, a malicious user is able to take action on behalf of the victim. Consider the following example. Mallory posts the following in a comment on a chat forum:

```
<img src="http://patsy-bank.com/transfer?amt=1000&to=mallory"/>
```

Of course, Patsy-Bank won't let just anyone request a transaction on behalf of any given account name. Users first need to authenticate with a password. However, once a user has authenticated, Patsy-Bank associates their session ID with an authenticated session state.

Q2.1 Explain what could happen when Alice visits the chat forum and views Mallory's comment.

Q2.2 Patsy-Bank decides to check that the `Referer` header contains patsy-bank.com. Will the attack above work? Why or why not?

Q2.3 Describe one way Mallory can modify her attack to always get around this check

Q2.4 Recall that the `Referer` header provides the full URL. HTTP additionally offers an `Origin` header which acts the same as the `Referer` but only includes the website domain, not the entire URL. Why might the `Origin` header be preferred?

Q2.5 Almost all browsers support an additional cookie field `SameSite`. When `SameSite=strict`, the browser will only send the cookie if the requested domain **and** origin domain correspond to the cookie's domain. Which CSRF attacks will this stop? Which ones won't it stop? Give one big drawback of setting `SameSite=strict`.

Question 3 ***Clickjacking*** **()**

In this question we'll investigate some of the click-jacking methods that have been used to target smartphone users.

Q3.1 In many smartphone browsers, the address bar containing the page's URL can be hidden when the user scrolls. What types of problems can this cause?

Q3.2 Smartphone users are used to notifications popping up over their browsers as texts and calls arrive. How can attackers use this to their advantage?

Q3.3 QR codes are used for various wide-ranging applications, for example: ordering at a restaurant, or providing a job link at a career fair. Can you think of any security vulnerabilities that might exist with the widespread use of QR codes?