1A. A self-signed certificate allows use of https in the sense of an encrypted conversation between the client and the server. However self-signed certificates do not allow you to ensure that the server is who they say they are, so identity is the issue here.

1B. In a man in the middle attack the adversary could self-sign with the same signature as the original server, and the client would interact with the adversary instead of the intended host.

1C. It is unwise to use https for the login page and then cookies for the rest of the pages. A hijacker could steal the session ID from the client after they login through the trusted https source instead of stealing the users password.

1CA. Self signed certificates wouldn’t be wise in this scheme either. Again self signed certificates ensure only encryption, and not identity. If a certificate is self signed someone can pretend to be the host from previous interactions.

1CB. A certificate signed by a trusted CA is the best way to ensure safety for web-traffic. This method provides encryption, and insures that you are talking to who you think you are.

# Problem 2

2A. If the attacker were to go into the dev console on chrome or Firefox and update their username in the form or in their session, then submit the form they could delete someone else’s account.

2B. If the attacker were to send a malicious link with the necessary parameters to the victim, then the victim’s browser would likely direct the victim to that link as an authenticated user. This is dangerous because the victim could unintentionally delete their account from clicking on a link.

2C. A XSS could be used in conjunction with the CSRF. The XSS would grab the login cookie while the CSRF ensured that the other login requirement was met, these two in conjunction would defeat the updated /deleteuser method.

3A. A random 64 bit integer prevents the basic version of a buffer overflow in that it is difficult to guess the 64 bit integer generated. In the majority of cases the canary will be overwritten with the wrong value, and execution will be halted.

3B. One drawback of determining the canary at compile time instead of run time is that at compile time it might be possible to determine the value of the canary, and use that value to overflow the buffer. With runtime determination of the canary it is more difficult to correctly determine the canary. 0 is a preferable static canary because it is a terminator, and would cause the overflow to stop.

3C. One way to beat a canary is to overwrite it with the same exact value, and allow it to be part of your buffer and pass the canary check. Another way is to edit the bubble up of the exception, and intercept/edit the exception stack at some point silencing the exception.

4A. An ethical worm is tricky, it might break systems with the security patch unintentionally. Beyond this people wouldn’t have the choice to opt-out of the worm, and remove the malware themselves. Even if the worm spread with minimal cost the web would still see increased bandwidth as well.

4B. One possibility time to use a active security counter attack would be in the case of a large bot net, like that of kelihos. In an attempt to disable the bot-net security researches used a sinkhole, and were able to redirect all of the infected machines into their grasp (even though the botnet was p2p).