Cross-Site Scripting

One day when checking your news and status updates on your favourite social network FaceSpace, you see that the string you searched for in a previous window is being displayed on the result page.

Having recently learned something about cross-site scripting, you decide to test searching for

```
<script>alert(42);</script>
```

1. What is the potential security issue that you are trying to test?

2. You then post what you had searched for instead a reply to a long thread of messages on FaceSpace "anti-faxxer" message board filled with people who do not like fax machines for some reason. What is the issue you are testing here? How it is different from the above one?

3. Suppose that both times the page generates a JavaScript pop-up alert. What are the security implications of this?

4. FaceSpace finds out about this vulnerability by accident through their comment quality division, who are now legally required to read all fax-machine-related comments posted by users to ensure accuracy. They fix it by removing all instances of <script> and </script>. Why isn't this sufficient? What is a better way to fix such vulnerabilities.

Cross-Site Not Scripting

Consider a simple web messaging system. You get a bunch of messages from other users, and the dashboard of the system shows all the messages ever sent to you. Its HTML looks like this:

```
<pre>
Eve: Alice! Where are you??
Amazon: Your account verification code is 347214
Eve: Hello????
Amazon: Thank you for your purchase of a <b>shrubbery</b> <img
src="https://www.amazon.ca/assets/thankyou.png>
</pre>
```

This site allows users to sent *arbitrary HTML code* that is just concatenated into the page, *unsanitized*. For the sake of argument, however, suppose that they have a magical technique to prevent *any kind of JavaScript* code from running. What message(s) could Eve send Alice so that when Alice views her dashboard, Eve is able to snoop on her other messages.