Chapter 1 Questions:

1. Why are an application's mechanisms for handling user access only as strong as the weakest of these components?

2. What is the difference between a session and a session token?

3. Why is it not always possible to use a whitelist-based approach to input validation?

4. You are attacking an application that implements an administrative function. You do not have any valid credentials to use the function. Why should you nevertheless pay close attention to it?

5. An input validation mechanism designed to block cross-site scripting attacks performs the following sequence of steps on an item of input:

   (a) Strip any `<script>` expressions that appear.

   (b) Truncate the input to 50 characters.

   (c) URL-decode the input.

   (d) If any items were deleted return to step 1.

   Can you bypass this validation mechanism to smuggle the following data past it?
   `"><scrpt>alert("foo")</script>`

1. Attackers are more likely to find the easiest of these vulnerabilities, which can then be used to bypass the user authentication.

2. A session is the stage of the application that a user is currently at. A session token is a string that identifies where the user if at so they can return.

3. Applications must be able to accept data for processing that does not meet any reasonable criteria.

4. Error messages could reveal important information about how the application is structured or the logic it uses.

5.