Questions & Answers for Web Security

Questions

Q1: What are the 3 properties of cryptographic hash functions?

Q2: Why is it not good to store user information as plaintext and what is a good alternative?

Q3: What are two alternatives to use in order to prevent SQL injection?

Q4: What is the difference between GET and POST methods and when do we use them?

Q5: What is/are the differences between *cookie* and *session*?

Q6: Why do we need to use HTTPS instead of HTTP?

Q7: What is a CSRF token?

Answers

A1: 1. There is no way to reverse the hash. You cannot get plaintext from cryptographic hash. 2. You cannot have the same output for two different inputs. 3. A small change in the input would create a big change in output.

A2: Because in that case everyone who has access to the database can see the password. Also, if the system is hacked and database is stolen, the passwords are in the open. That is why a better way to store password is adding a random salt value to the password and storing them by hashing that total value.

A3: Using ORM and prepared / parameterized SQL.

A4: If we want to create new resources, we use POST. If we want to read or retrieve data, we use GET. We should never use GET to create or modify data on the server.

A5: Cookies are client-side files, they contain information but they are stored in the client. Sessions on the other hand are stored in the server. The amount of data stored in the cookies are limited, whereas you can store a lot of data in sessions.

A6: Because HTTPS is the secure version of the HTTP. In HTTP, the communication between the client and server is not encrypted and is in plaintext. Anyone who can get in the middle of the communication can retrieve and read the data. There is no privacy or secrecy. HTTPS encrypts packets so that even if someone can retrieve the packet, they cannot decrypt and therefore get information on it.

A7: It is a secret value in every form so that the server can validate it. Unless you know the token, it is impossible to submit a form from a remote location. It is an additional secret value in the form to eliminate remote submits.