HOMEWORK 5

RÜMEYSA TÜRKER

1 – What are Authentication and Authorization ?

Authentication is the process of identifying users and validating who they claim to be. One of the most common and obvious factors to authenticate identity is a password. If the user name matches the password credential, it means the identity is valid, and the system grants access to the user.

Authorization is a process by which a server determines if the client has permission to use a resource or access a file.Authorization is usually coupled with authentication so that the server has some concept of who the client is that is requesting access.The type of authentication required for authorization may vary; passwords may be required in some cases but not in others.

2 - What is Hashing in Spring Security ?

Hashing is the process of generating a string, or hash, from a given message using a mathematical function known as a cryptographic hash function.

While there are several hash functions out there, those tailored to hashing passwords need to have four main properties to be secure:

- It should be deterministic: the same message processed by the same hash function should always produce the same hash
- It's not reversible: it's impractical to generate a message from its hash
- It has high entropy: a small change to a message should produce a vastly different hash
- And it resists collisions: two different messages should not produce the same hash

A hash function that has all four properties is a strong candidate for password hashing since together they dramatically increase the difficulty in reverse-engineering the password from the hash.

3 - What is Salting and why do we use the process of Salting ?

Salting involves adding random data before it is put through a cryptographic hash function. It's mostly used to keep passwords safe during storage, but it can also be used with other types of data.

Salts are used to safeguard passwords in storage. Historically, only a cryptographic hash function of the password was stored on a system, but over time, additional safeguards were developed to protect against duplicate or common passwords being identifiable (as their hashes are identical).

4 - What is "intercept-url" pattern ?

The pattern attribute on the security:intercept-url tag uses Ant paths. Under this syntax . has no meaning except for a literal . So the pattern is most likely not recognized by Spring security and ignored.

5 - What do you mean by session management in Spring Security ?

Session management refers to the process of securely handling multiple requests to a web-based application or service from a single user or entity. Websites and browsers use HTTP to communicate, and a session is a series of HTTP requests and transactions initiated by the same user. Typically, a session is started when a user authenticates their identity using a password or another authentication protocol. Session management involves the sharing of secrets with authenticated users, and as such, secure cryptographic network communications are essential to maintaining session management security.

Session Management is very crucial part for the Spring Security because if session is not managed properly, then security of data is directly impacted.

6 – Why we need Exception Handling ?

Java exception handling is important because it helps maintain the normal, desired flow of the program even when unexpected events occur. If Java exceptions are not handled, programs may crash or requests may fail. This can be very frustrating for customers and if it happens repeatedly, you could lose those customers.

7 - Explain what is AuthenticationManager in Spring security ?

AuthenticationManager is the main strategy interface for authentication.It is a core interface that spring security uses for the authentication process. It has only one method authenticate which when implemented in a class that implements an Authentication Manager has all the logic for authenticating a user request. The authenticate method takes an “Authentication” object as its parameter and returns an “Authentication” object on successful authentication of the user or else we can have an exception thrown indicating that the user is not authenticated.

8 - What is Spring Security Filter Chain ?

Spring Security maintains a filter chain internally where each of the filters has a particular responsibility and filters are added or removed from the configuration depending on which services are required. The ordering of the filters is important as there are dependencies between them. If you have been using namespace configuration, then the filters are automatically configured for you and you don't have to define any Spring beans explicitly but here may be times when you want full control over the security filter chain, either because you are using features which aren't supported in the namespace, or you are using your own customized versions of classes.

9 – What are the differences between OAuth2 and JWT ?

OAuth 2.0 defines a protocol, i.e. specifies how tokens are transferred, JWT defines a token format.

OAuth 2.0 and "JWT authentication" have similar appearance when it comes to the (2nd) stage where the Client presents the token to the Resource Server: the token is passed in a header.

But "JWT authentication" is not a standard and does not specify how the Client obtains the token in the first place (the 1st stage). That is where the perceived complexity of OAuth comes from: it also defines various ways in which the Client can obtain an access token from something that is called an Authorization Server.

So the real difference is that JWT is just a token format, OAuth 2.0 is a protocol (that may use a JWT as a token format).

10 - What is method security and why do we need it ?

Spring method security allows us to support / add authorization supports at the method level. On a high level, we can configure which roles are allowed to access what method within the same service class.

The @Secured annotation is used to specify the list of roles on a method. This allows us to provide access to a specific method in case the user has a role.

11 – What Proxy means and how and where can be used ?

Proxy servers act as intermediaries between client applications and other servers. In an enterprise setting, we often use them to help provide control over the content that users consume, usually across network boundaries.

A proxy server acts as a gateway between you and the internet. It’s an intermediary server separating end users from the websites they browse. Proxy servers provide varying levels of functionality, security, and privacy depending on your use case, needs, or company policy.

12 – What is Wrapper Class and where can be used ?

Wrapper classes are used to convert any data type into an object. The primitive data types are not objects; they do not belong to any class; they are defined in the language itself. Sometimes, it is required to convert data types into objects in Java language.

13 – What is SSL ? What is TLS ? What is the difference ? How can we use them ?

SSL stands for Secure Sockets Layer and, in short, it's the standard technology for keeping an internet connection secure and safeguarding any sensitive data that is being sent between two systems, preventing criminals from reading and modifying any information transferred, including potential personal details

TLS (Transport Layer Security) is just an updated, more secure, version of SSL. We still refer to our security certificates as SSL because it is a more commonly used term, but when you are buying SSL from DigiCert you are actually buying the most up to date TLS certificates with the option of ECC, RSA or DSA encryption.

SSL	**TLS**
SSL stands for “Secure Socket Layer.”	TLS stands for “Transport Layer Security.”
Netscape developed the first version of SSL in 1995.	The first version of TLS was developed by the Internet Engineering Taskforce (IETF) in 1999.
SSL is a cryptographic protocol that uses explicit connections to establish secure communication between web server and client.	TLS is also a cryptographic protocol that provides secure communication between web server and client via implicit connections. It’s the successor of SSL protocol.
Three versions of SSL have been released: SSL 1.0, 2.0, and 3.0.	Four versions of TLS have been released: TLS 1.0, 1.1, 1.2, and 1.3.

All versions of SSL have been found vulnerable, and they all have been deprecated.	TLS 1.0 and 1.1 have been “broken” and are deprecated as of March 2020. TLS 1.2 is the most widely deployed protocol version.

14 - Why do you need the intercept-url ?

The <intercept-url> element defines a pattern which is matched against the URLs of incoming requests using an ant path style syntax.

Most web applications using Spring Security only have a couple of intercept-urls because they only have very basic security requirements. You need to have unauthenticated access to the login and login-error screens and usually some aspect of the public site, so that can be a few URL patterns