HW#5

1 - What are Authentication and Authorization?

Authentication is the process of verifying the identity of user who request to access to a system. It is the first step in any security process. 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 accessibility of someone. It determines what resources a user can access. It is about offering full or partial access rights to resources like database or other critical information. In secure environments, authorization must always follow authentication. Users should first prove that their identities are genuine before an organization's administrators grant them access to the requested resources.

2 - What is Hashing in Spring Security?

Hashing is the process of converting an input of variable length to a fixed size array of numbers and letters using a mathematical formula. End of this process a result is being produced. And this result is called hash.

Hashing works in one direction only. For a given piece of data, we will always get the same hash but we can't turn a hash back into its original data. So, when we want to store some information without being able to read it, like user passwords, we use hashing. Hashing process changes passwords into hashes and store them like that. If we want to read those data we can't because hashed data can't be hash back. When a user try to login, system hashes the password which user enters during login and compare the output with the stored hashed value. If the values match then the user has entered the correct password and is allowed to proceed.

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

Salting is an additional step during hashing passwords. This step adds an additional value to the end of the password and changes the hash value produced. That additional value is called salt.

Salting adds a layer of security to the hashing process especially against brute force attacks. A brute force attack is where a computer attempts to figure out a password with every possible combination of letters and numbers until the password is found. With salting we add a value to the password and hash it after that. Hashing with an additional value creates a completely different hash and it gets really difficult to figure out password because the produced hash is too different from original passwords hash. Of course, for a successful salting process, attackers shouldn't know the salt value.

4 - What is “intercept-url” pattern?

The *<intercept-url>* element defines a pattern which is matched against the URLs of incoming requests using an ant path style syntax. It is used with “*access*” attribute and this attribute defines the access requirements for requests matching the given pattern.

5 - What does it mean by session management in Spring Security?

When a user visits a website, a session is made containing multiple requests and responses over HTTP. Spring security session management manages sessions between the web application and the user. It is a mechanism to detect session timeouts. Spring security session management provides us to be able to take some actions which we specified after session timeouts.

By default, Spring security creates session when required. But Spring security provides us option to configure when the session will be created. So with the help of session management we can define the exact moment when a session is going to be created.

6 - Why do we need Exception Handling?

Exceptions are events that disrupt the normal flow of a program’s execution. Exception handling allows us to handle errors or other problems in a program that would cause the program to crash while execution. With exception handling we can create a logic that whenever a known or an unknown exceptional condition occur, we can handle them. So we can say that exception handling mechanism protects the program from being stuck or terminated.

Java uses the *try-catch* construct to handle exceptions. There are two types of exceptions: runtime exception and compile time exception.

7 - What is AuthenticationManager in Spring security?

AuthenticationManager is the main strategy interface that defines how Spring Security’s Filters perform authentication. It processes an Authentication request. *AuthenticationManager* has only one method which is called “*authenticate()*”.

AuthenticationManager can perform one of three actions with its “*authenticate()*” method. First option is to return an *Authentication* instance (with *authenticated=true*) if it can verify that the input represents a valid principal. Second option is to throw an *AuthenticationException* if it believes that the input represents an invalid principal. And the last option is to return *null* if it cannot decide.

8 - What is Spring Security Filter Chain?

Spring Security is based on a chain of servlet filters. When a request is being send for a resource, spring application container creates a filter chain to process that incoming request. Spring Security maintains the filter chain internally where each of the filters has a specific 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.

9 - What are the differences between OAuth2 and JWT?

JWT is a technology for API authentication and server-to-server authorization. JWT is a JSON based format of a security token which is basically a base64 url-encoded string which is used as a means of transferring. So technically it is a token format. It secures the content between two applications, especially request data in Web APIs.

OAuth is an open standard for authorization and anyone can implement it. OAuth is a standard that apps can use to provide client applications with “secure delegated access”. With OAuth we can log into third party websites with our Google, Facebook, Twitter or Microsoft accounts without having the necessity to provide our passwords. OAuth2 is the version 2 of the OAuth protocol.

So the main difference between JWT and OAuth2 is that JWT is just a token format but OAuth 2.0 is a protocol which can use JWT as a token. JWT is a JSON based security token for API authentication. OAuth2 is a standard set of steps for obtaining a token. It specifies the flows and standards under which authorization token exchanges should occur.

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

Spring method security means configuring which roles are allowed to access what method within the same service class. Spring method security allows us to add authorization security to individual methods within our service layer. We can say that it is an additional layer of security. It is useful especially in complex applications. If we have a complex application where some service methods can be called from different controllers and we want to make sure that we did not fail to restrict an access, method security helps by ensuring that only valid users can do certain business actions.

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

Proxy is a surrogate object or placeholder for another object to control access to it. The proxy sits in between the caller of an object and the real object itself. It can decide to prevent the target object from being invoked or do something before the target object is invoked. In other words, proxies can be used as stand-ins for real objects to apply extra behavior to those objects–be it security-related behavior, caching or maybe performance measurements.

Many modern frameworks use proxies to realize functionality that would not have been possible otherwise. Many object-relational mappers use proxies to implement behavior that prevents data from being loaded until it is actually really needed. Spring also uses proxies to realize some of its functionality such as its remoting facilities, its transaction management facilities and the AOP framework.

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

Wrapper classes provide us a way to use primitive data types as objects. The automatic conversion of primitive data type into its corresponding wrapper class is known as autoboxing. The automatic conversion of wrapper type into its corresponding primitive type is known as unboxing.

Java Wrapper classes wrap the primitive data types, that is why they are known as wrapper classes. We can also create a class which wraps a primitive data type. So we can create a custom wrapper class in Java.

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

SSL (Secure Socket Layer) and TLS (Transport Layer Security) are cryptographic protocols that are used to secure an internet connection by encrypting data sent between two systems (servers, machines and applications operating over a network). They prevent hackers from seeing or stealing any information transferred.

SSL was developed in the mid-1990s. TLS is an updated, more secure version of SSL and was released in 1999. We still refer to our security certificates as ‘SSL certificate’ because it’s a more common term but SSL has been completely deprecated and no modern systems support SSL anymore.

Encryption is necessary in order to communicate securely over the internet. TLS serves to provide end-to-end encryption for all data transmitted from one point to another and uses cryptography to ensure that only the two transacting bodies are capable of reading this information. Connections that are secured by TLS (or SSL) will indicate their secure status by displaying HTTPS (Hypertext Transfer Protocol Secure) in the address bar of web browsers.