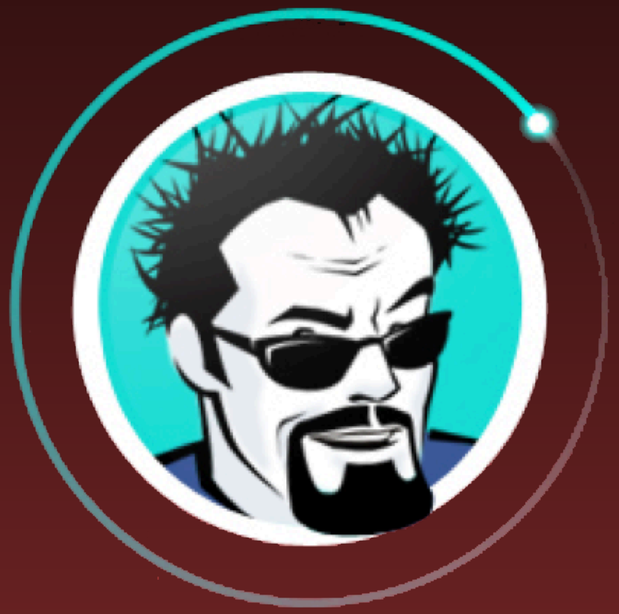


Spring Security

Beginner to Guru

Introduction to Spring Security

Many Levels of Security

- In the context of computing, there are many levels of security:
  - **Hardware Security** - prevent unauthorized code execution
  - **Operating System** - access to the computer and actions you can take
  - **Database** - access to the database and actions you can take
  - **Message Brokers** - read / write access to message queues
  - **Network Security** - wifi, VPCs, etc
  - **Application Security** - Access to the application and actions within application

Spring Security

- Spring Security focuses on **Application Security**
  - Spring Security does not address other levels of security
- Application Security focuses on who can do what within the context of an application
- Spring Security provides:
  - Protection from common security exploits
  - Integration with external security products, such as LDAP
  - Provides utilities for password encoding

Application Security Key Terms

- **Identity** - A unique actor, typically an individual aka user
- **Credentials** - Usually a user id and password
- **Authentication** - Is how the application verifies the identity of the requestor
  - Spring Security has a variety of methods for Authentication
  - Typically the user provides credentials, which are validated
- **Authorization** - Can a user perform an action?
  - Using the user's identity, Spring Security determines if they are authorized to perform action

Authentication Providers

- **Authentication Providers** - Verify users identities
- Authentication Providers supported by Spring Security:
  - In Memory
  - JDBC / Database
  - Custom
  - LDAP / Active Directory
  - Keycloak
  - ACL (Access Control List)
  - OpenID
  - CAS

Password Storage

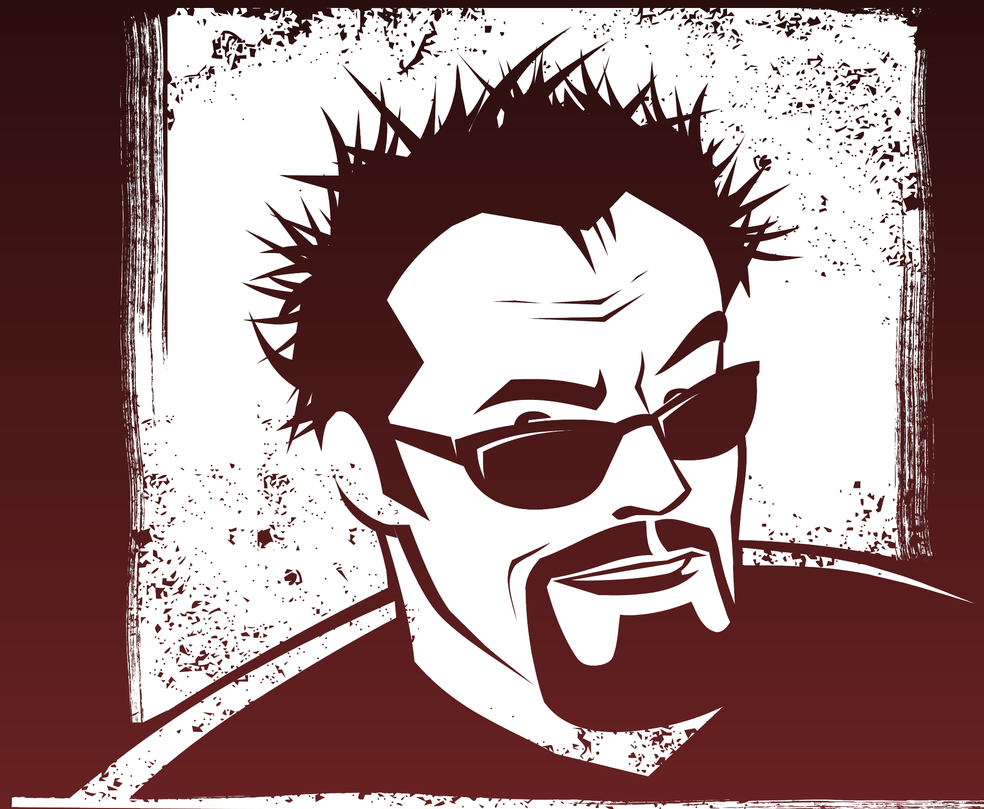
- Spring Security supports a variety of methods to store and verify passwords
  - **NoOp** Password Encoder - plain text, not recommended - for legacy systems
  - **BCrypt** - uses bcrypt password hashing
  - **Argon2** - Uses Argon2 algorithm
  - **Pbkdf2** - Uses PBKDF2 algorithm
  - **SCrypt** - Uses scrypt algorithm
  - **Custom** - Roll your own? Not recommended!

Spring Security Modules

- **Core** - Core modules of Spring Security
- **Remoting** - Only needed for support of RMI operations
- **Web** - Support of web applications
- **Config** - Provides support for XML and Java configuration
- **LDAP** - for integration with LDAP identity providers
- **OAuth 2.0 Core** - Core of OAuth 2.0 Authorization and OpenID
- **OAuth 2.0 Client** - Client support for OAuth 2.0 and OpenID clients

Spring Security Modules

- **OAuth 2.0 JOSE** - Provides support for JOSE (Javascript Object Signing and Encryption)
- **OAuth 2.0 Resource Server** - Support for OAuth 2.0 Resource Servers
- **ACL** - Support for Access Control Lists
- **CAS** - Support for Central Authentication Service
- **OpenID** - Authenticate users with external OpenID server
- **Test** - Testing Support for Spring Security



SPRING FRAMEWORK

GURU