Course code:	**SPRSEC**
Course title:	**Implementation of security mechanisms in Web applications using Spring Security, Apache Shiro, JBoss Picketlink**
Days:	3

Description:

Course intended for:

The training is intended for programmers wanting to learn and understand the functioning of Spring Security 3 quickly and gain a general understanding of alternative possibilities of implementation of security mechanisms in Web applications and in the JEE environment

Course objective:

The training objective is to get familiar with programming and configuration techniques allowing for implementation of security mechanisms in Web applications using Spring Security 3.

The training in its basic format provides a detailed discussion of capabilities of Spring Security 3. The remaining solutions have been discussed only theoretically for the sake of comparison. On request, the training curriculum can be adjusted – the technology discussed in detail can be e.g. Apache Shiro, while other issues (that is, Spring Security 3, JBoss Picketlink) are a complementation only.

- Security in the world of JEE
- JEE application security and Web application security
- Discussing of the principles of functioning of Spring Security with substantial emphasis on the architecture of the solution and correlations between its components (objects, services) – necessary to understand this solution
- Practical use of Spring Security mechanisms delivered
- broadening of the scope of Spring Security mechanisms available for the purpose of implementation of own requirements
- How to speed up the process of testing of the security mechanisms implemented through tests overriding the container?

Course strengths:

Spring Security is considered to be one of the best solutions, allowing for implementation of security mechanisms in Web applications. On the other hand, many believe this solution to be too complicated, as it required much time for understanding of the underlying principles of this mechanism. This training addresses these problems, providing specific information that allows the users to get familiar with the world of Spring Security 3. The training starts with a theoretical introduction, during which key concepts and dependencies are discussed that are necessary to be able to „control" this solution. The key objects and services and correlations between them are presented. Apart from theory, in the further part of the training, the issues discussed are used in practice. Thanks to this, in a relatively short period of time, the user will be able to acquire a much broader scope of knowledge and skills than it would be possible for the participants on their own.

Requirements:

The training participants are required to have the Java language programming skills (to be learned at the course J/JP), to be familiar with the basic issues of the Spring framework (to be learned at the course J/SPRING) and with the basic concepts associated with Web application programming (to be learned at the course J/WEB2b).

Course parameters:

3*8 hours (3*7 net hours) of lectures and workshops (with a visible emphasis on workshops).

Group size: no more than 8-10 participants.

Course curriculum:

1. Security of JEE applications and security of Web applications – introduction

    I. Security from the perspective of the JEE standard (authentication, authorization, data integrity, data transmission security)

    II. Basic terms (credentials, principal, realm, session etc.)

    III. Discussing of available authentication mechanisms in Web modules

        i. HTTP Basic Authentication: BASIC

        ii. Digest Authentication: DIGEST

        iii. HTTPS Client Authentication: CLIENT-CERT

iv. Form-Based Authentication: FORM

IV. Security of data transmission (transport security)

i. 2 available levels: Confidential and Integral

V. Authorization:

i. Declarative: use of available annotations (e.g. @RolesAllowed, @PermitAll , @DeclareRoles, @RunAs , @DenyAll, @ServletSecurity)

ii. Programming: use of methods (getRemoteUser(), isUserInRole(), getUserPrincipal(), getAuthType() , login(), logout(), getScheme())

VI. Discussion of specific characteristics of EJB module protection

i. Authentication and authorization in the area of session beans and entities

ii. EJB Deployment Descriptors and dependence on the selected application server

2. A review of the available techniques/ solutions allowing for programming of security mechanisms in the JEE application:

I. Security based on application server mechanisms

II. Overriding of application server in implementation of security mechanisms

III. JAAS

IV. Spring Security

V. Apache Shiro

VI. PicketBox

3. Spring Security – discussing of key concepts and their correlations

I. SecurityContextHolder, SecurityContext

II. Authentication, AuthenticationEntryPoint, UserDetailsService, UserDetails, Principal, Credentials, GrantedAuthority

III. Immediately available implementations of UserDetailsService e.g.

InMemoryDaoImpl, JdbcDaoImpl

IV. AuthenticationManager, UsernamePasswordAuthenticationToken, ProviderManager

V. AuthenticationProviders with the available implementations, e.g. DaoAuthenticationProvider, LdapAuthenticationProvider, CasAuthenticationProvider

VI. Password Encoders (hash, salt)

VII. AOP and authorization mechanisms

VIII. AccessDecisionManager, Secure Objects, AccessDecisionVoter (e.g. RoleVoter), RunAsManager, AfterInvocationManager, AbstractSecurityInterceptor

IX. Filters in the security process e.g. DelegatingFilterProxy, FilterChainProxy, BasicAuthenticationFilter, UsernamePasswordAuthenticationFilter, RememberMeAuthenticationFilter, FormLoginFilter, ExceptionTranslationFilter, ConcurrentSessionFilter, SecurityContextPersistenceFilter

    i. Overriding of filter chain (filters = "none")

    ii. The impact of order, in which filters were declared, on the authentication and authorization process

    iii. Own filters

X. Configuration using the name space

    i. Benefits of declarative configuration

    ii. Discussing of name space components that exert impact on configuration (Web/HTTP, Authentication Manager, Authentication Providers, UserDetailsService, AccessDecisionManager, BusinessObject)

    iii. auto-config, form-login, logout attributes

XI. The model of exceptions in Spring Security 3

    i. AuthenticationException and available subclasses e.g. BadCredentialsException, UsernameNotFoundException

    ii. AccountStatusException as a subclass of AuthenticationException but also as one of the most significant classes supporting management of „invalid" user accounts. (subclasses e.g. AccountExpiredException, LockedException, DisabledException or CredentialsExpiredException)

iii. AccessDeniedException with subclass AuthorizationServiceException

4. Authentication in practice

I. Implementation of the simple authentication mechanism, using name space configuration mechanisms

II. Implementation of the simple authentication mechanism in "in-memory" mode using mechanisms available in Spring Security 3 (e.g. InMemoryDaoImpl)

III. A standard, predefined data model representing user accounts and their roles

IV. Implementation of the simple authentication mechanism in the "jdbc" mode using the mechanisms available in Spring Security 3 (e.g. JdbcDaoImpl) and a predefined data model

V. Authentication in the „Remember-me" mode

5. Authorization in practice

I. Authorization using RoleVoter and AuthenticationVoter

II. Protection of business method requests

III. Using of expressions in access control (e.g. hasRole, principal, isAuthenticated, isFullyAuthenticated)

IV. Annotations @PreAuthorize, @PreFilter, @PostAuthorize and @PostFilter

6. Spring Security – advanced issues

I. Testing outside container – how to speed up the process of testing of the solutions implemented?

II. Implementation of own logic and data model, which will support non-standard requirements (e.g. having no predefined implementation in Spring Security) with regard to user authentication and authorization

III. Practical use of ACL (Access Control List)

IV. Use of SSL

V. User session management

VI. Control of Web page content via the available tag library (tags: authorize, authentication, accesscontrollist)

7. What else is worth knowing (a theoretical outline)

I. LDAP Authentication

II. JAAS Provider

III. CAS Authentication

IV. X.509 Authentication

V. Support of OpenID

VI. Projects enhancing the capabilities of Spring Security

i. The Spring Crypto module and its support for (symmetric) encoding, key generation or password coding

ii. The Spring Security Extension project and its support for SSO, integration with Kerberos and SAML2

iii. OAuth for Spring Security Project and its support for OAuth

iv. Summary