

Formation Sécurité Java/JEE

Comprendre et mettre en place une politique de sécurité sur une application Java/JEE

Référence : SECURITE-JEE-02

Durée : 2 jour(s)

Objectifs

- Comprendre les besoins de sécurisation liés aux architectures des Systèmes d'information
- Comprendre les moyens de sécurisation disponibles et en particulier les PKI
- Savoir sécuriser une application Java/JEE de manière efficace

Répartition: 40% Théorie, 60% Pratique

Public: Architecte, Développeur, Chef de projet

Pré-requis: Connaissance de Java

Programme

Présentation

- La sécurité, une vieille histoire...
- Bob, Alice et Charly

Problématiques de sécurisation

- Identification
- Authentification
- Autorisation
- Confidentialité
- Non-répudiation

Techniques de sécurisation

- Chiffrement (DES, AES, ...)
- Code de hachage
- Signature (MD5, ...)

JCE : Java Cryptography Extension

- Mise en œuvre du chiffrement et du déchiffrement
- Mise en œuvre de la signature
- Configuration et choix des Security Provider

PKI : les Infrastructures à clef publique

- Différences entre une Clef, une Bi-Clef et un Certificat
- Illustration d'un envoi d'email avec thunderbird
- Les acteurs d'une PKI
- Autorités de Certification
- Exemples : Entrust, VeriSign, Thawte
- Créer votre propre AC
- Listes de révocation
- Où stocker toutes ces clefs ?

Keytool : le magasin de clefs Java

- Génération, manipulation, export et import de clefs et de certificats

SSL

- Le ou les finalités de SSL
- SSL simple / SSL mutuelle
- Procédure de Handshake
- Présentation de TLS
- HTTP + SSL = HTTPS
- Les Magasins de certificats de Internet Explorer et Firefox

JSSE : l'API SSL du monde Java

- Présentation de Java Secure Socket Extension
- Manipulation de certificats X.509
- Sécurisation d'un échange client/serveur

JAAS : Java Authentication and Authorization Service

- Présentation du principe et des acteurs JAAS
- Notion de Principal et de Subject
- Présentation de modules de Login classiques
- Exemples: Windows, LDAP, SGBD, ...

Sécurité de la JVM

- Sécurité liée au classloader
- Classloader par défaut
- Politique de chargement des classes
- Créer votre propre classloader
- Garantir l'intégrité du code par signature
- Inconvénients du ByteCode : décompilation et obfuscation de code
- Apprendre à activer le SecurityManager
- Définir des politiques d'accès grâce aux JavaPolicy (*.policy)

- Mettre en œuvre les SecurityPermissions
- Présentation du Bac à sable (sandbox)
- Cas des Applets
- Cas des applications JavaWebStart

Sécurité d'une application JEE




- Une sécurité basée sur les rôles (security role based)
- Notion de Realm : JDBC, LDAP, XML, ...
- Sécurité déclarative par Security Constraints dans les descripteurs de déploiement
- Sécurité programmatique par l'utilisation d'API dans le code applicatif
- Sécurisation des EJB et des Applications Web
- Exemple de configuration du container web Tomcat
- Problématique d'accès aux autres acteurs : Exemple avec la base de données

Moyens d'authentification standards

Introduction à la Sécurité matérielle

- Hub et Switch
- Firewall et DMZ
- Ouverture de ports
- Système d'exploitation
- Proxy / Reverse Proxy
- Réseaux privés et VPN

OFFERT EN INTER-ENTREPRISE

-  Le petit déjeuner croissants, jus d'orange, café)
-  Le déjeuner
-  Une qualification téléphonique si nécessaire avec l'un de nos consultants

Tel: +33(0)1 45 26 19 15
Fax : +33(0)1 75 43 49 92
Email : training@zenika.com

Auteur du cours



Laurent Delvaux est formateur certifié, ainsi qu'évangéliste BIRT et met régulièrement en œuvre cette technologie sur des projets d'envergures. Scrum Master et agiliste convaincu, il aide les équipes de développement à s'auto-structurer et à augmenter leur productivité à l'aide de Scrum. A ce titre, il est membre du bureau du French Scrum User Group.



zenika
ARCHITECTURE INFORMATIQUE