

Table des matières

Avant-propos	11
Chapitre 1. Pourquoi sécuriser les applications web ?	15
1.1. Qu'est-ce qu'une application web ?	15
1.1.1. Internet, un réseau mondial	15
1.1.2. Les programmes avant le web	16
1.1.3. Une technologie web qui s'impose petit à petit dans les applications	17
1.1.4. Des échanges basés sur la confiance	18
1.1.5. Une mauvaise idée : l'intranet serait naturellement sûr	19
1.2. Qu'est-ce que la sécurité en informatique ?	20
1.2.1. Une sécurité qui repose sur de nombreuses briques	20
1.2.2. Des besoins et des niveaux de sécurité différents	22
1.3. Quelques exemples de dommages occasionnés par des manquements à la sécurité	23
1.3.1. Rien n'est jamais acquis...	26
1.3.2. Une application bien structurée et plus facile à protéger	27
1.3.3. Une sécurité seulement globale	27
1.3.4. Quelle sécurité pour les applications de type client lourd ?	28
Chapitre 2. Estimer le risque	31
2.1. Qu'est-ce que le risque ?	31
2.2. Comment se protéger du risque ?	32
2.3. Déterminer la cible	33
2.4. Déterminer l'impact	34
2.4.1. La confidentialité	34
2.4.2. L'intégrité	35

- 2.4.3. La disponibilité 37
- 2.4.4. Déterminer le niveau de risque associé 38
- 2.5. Quelles causes ou quels scénarios prendre en compte ? 40
 - 2.5.1. Les exigences ASVS 41
 - 2.5.2. Déterminer les causes à prendre en compte
et leur occurrence potentielle 42
 - 2.5.3. Déterminer le niveau d'exigences à prendre en compte 43
- 2.6. Comment mener l'étude dans l'entreprise ? 43

Chapitre 3. Le chiffrement et la configuration du serveur web 45

- 3.1. Les différents serveurs web 45
- 3.2. Quelques notions sur le chiffrement 46
 - 3.2.1. Le chiffrement symétrique 46
 - 3.2.2. Le calcul d'empreintes et le salage de mots de passe 47
 - 3.2.3. Le chiffrement asymétrique 49
 - 3.2.4. Quelle longueur pour les clés de chiffrement ? 51
 - 3.2.5. Les certificats numériques et la chaîne de certification 52
- 3.3. Générer et manipuler des certificats de chiffrement 54
 - 3.3.1. La bibliothèque OpenSSL 54
 - 3.3.2. Les différents types de certificats 54
 - 3.3.3. Générer les certificats 54
 - 3.3.3.1. Créer le certificat racine 55
 - 3.3.3.2. Créer les clés privée et publique du serveur 55
 - 3.3.3.3. Créer la requête de certificat qui sera validée
par l'autorité racine 55
 - 3.3.3.4. Signer le certificat localement 56
 - 3.3.3.5. Faire signer le certificat par une autorité d'enregistrement 57
 - 3.3.3.6. Créer un certificat autosigné sans utiliser d'autorité
d'enregistrement 57
 - 3.3.4. Où stocker les clés et certificats ? 57
 - 3.3.5. Quelques commandes pour consulter les clés et les certificats 58
- 3.4. Implémenter le protocole HTTPS 60
 - 3.4.1. Comprendre le protocole HTTPS 60
 - 3.4.2. Implémenter le protocole HTTPS 62
 - 3.4.3. Tester la chaîne SSL 63
- 3.5. Rendre plus sûr le serveur Apache 64
 - 3.5.1. S'assurer que le serveur hébergeant Apache dispose
de mises à jour de sécurité 64
 - 3.5.2. Interdire l'utilisation de protocoles de sécurité faibles 65
 - 3.5.3. Empêcher l'envoi en rafale de requêtes 66
 - 3.5.4. Mettre en place un filtre de requêtes 67
 - 3.5.5. Autoriser la réécriture des en-têtes de pages 69
 - 3.5.6. Autoriser l'utilisation des fichiers *.htaccess* 69

3.5.7. Masquer les versions d'Apache et de PHP	70
3.6. Conclusion	71
Chapitre 4. Les menaces et comment s'en protéger	73
4.1. Les menaces induites par l'environnement web	73
4.1.1. Limiter les types de requêtes autorisés	74
4.1.2. Interdire la navigation dans l'arborescence du site	75
4.1.3. Limiter le risque de vol du <i>cookie</i> de session	75
4.1.4. Masquer les messages d'erreur	76
4.1.5. Demander aux navigateurs de mettre en place des mesures de protection	76
4.2. Le top 10 des attaques en 2013	78
4.2.1. L'injection de code	78
4.2.2. Outrepasser l'identification et s'approprier la session	85
4.2.3. Exécuter du code pour rediriger vers un autre site, ou <i>Cross Site Scripting</i> (XSS)	88
4.2.4. Références directes non sécurisées à un objet	90
4.2.5. Mauvaise configuration de la sécurité de l'application et de son environnement	94
4.2.6. Exposition de données sensibles	94
4.2.7. Absence de contrôles de niveau d'accès pour accéder à certaines fonctions	96
4.2.8. Faire exécuter une instruction légitime à un utilisateur à son insu	97
4.2.9. Utiliser des composants connus comme étant vulnérables	99
4.2.10. Ne pas valider les redirections	99
4.3. Quelques parades complémentaires...	99
4.3.1. Vérifier l'encodage UTF-8	99
4.3.2. Analyser les documents « télé-versés » avec un antivirus	101
4.3.2.1. Analyser avec un module complémentaire	101
4.3.2.2. Utiliser le programme <i>clamscan</i>	104
4.3.2.3. Intégrer les deux méthodes	105
4.3.3. Interdire le stockage du <i>login</i> et du mot de passe par le navigateur	106
4.3.4. Chiffrer les accès à la base de données	109
4.3.4.1. Configurer PostgreSQL pour activer le support SSL	109
4.3.4.2. Configurer MySQL pour activer le support SSL	110
4.3.4.3. Activer la connexion en mode chiffré dans PDO	111
4.4. Mettre en place un contrôleur de ressources	112
4.4.1. Gérer la connexion des utilisateurs	113
4.4.2. Surveiller le comportement	116
4.4.3. Gérer les alertes	120

Chapitre 5. Gérer l'identification des utilisateurs et leur affecter des droits	123
5.1. Gérer l'identification des utilisateurs	123
5.1.1. Gérer les comptes dans la base de données	124
5.1.1.1. Réinitialiser un mot de passe	125
5.1.1.2. Contrôler la complexité du mot de passe	130
5.1.2. Verrouiller les mots de passe	134
5.1.3. Récupérer l'identification auprès de l'annuaire de l'entreprise	135
5.1.4. Déléguer l'identification à un serveur CAS	137
5.1.4.1. Principe du CAS	138
5.1.4.2. Implémenter l'identification à partir d'un serveur CAS	143
5.1.5. Aller plus loin avec le CAS : fédération d'identités, Shibboleth	145
5.1.6. Gérer l'identification en mode déconnecté avec un stockage en base de données	146
5.1.7. Gérer l'identification avec un jeton chiffré avec des clés asymétriques	152
5.1.8. Créer des jetons avec le protocole JWT	160
5.1.9. Utiliser le protocole OAuth pour générer des jetons	163
5.2. Gérer les droits	164
5.2.1. Que protéger ?	164
5.2.2. Gérer les droits des utilisateurs à partir des groupes de l'annuaire LDAP	169
5.2.3. Gérer les droits des utilisateurs à partir de groupes définis dans l'application	171
5.3. Conclusion	175
 Chapitre 6. Structurer l'application avec le modèle MVC	 177
6.1. Pourquoi structurer l'application ?	177
6.2. Qu'est-ce que le modèle MVC ?	178
6.2.1. Le modèle	179
6.2.2. La vue	180
6.2.3. Le contrôleur	183
6.2.3.1. Les contrôleurs liés	183
6.2.3.2. Les contrôleurs hérités	185
6.2.3.3. Les contrôleurs mono-objets	186
6.3. Conclusion	186
 Chapitre 7. Mettre en place une plate-forme technique adaptée et tester l'application	 189
7.1. Concevoir une architecture technique adaptée	189
7.1.1. Intégrer la sécurité dès le début des travaux	189
7.1.2. Utiliser des gestionnaires de code comme GIT	190

7.1.3. Utiliser des logiciels pour dessiner la base de données	191
7.1.4. Mettre en place des architectures séparées pour le développement et la production	191
7.2. Tester la sécurité de l'application	193
7.2.1. Analyser les vulnérabilités avec ZAP Proxy	193
7.2.1.1. Mettre en place une plate-forme de test	195
7.2.1.2. Configurer le navigateur	196
7.2.1.3. Lancer la découverte automatique de l'arborescence	196
7.2.1.4. Parcourir l'application	196
7.2.1.5. Désactiver l'attaque pour les pages de connexion et de déconnexion	197
7.2.1.6. Lancer l'attaque	197
7.2.1.7. Analyser les résultats	197
7.2.2. Certifier l'application	198
7.2.3. Rédiger les documents d'implémentation	198
7.3. Que faire si l'implémentation de la sécurité semble une tâche insurmontable ?	199
Bibliographie	201
Index	205