# Joomla! et la sécurité Web

THIERRY MEYER

thierry.meyer@tm-consultants.fr

# Introduction

# A propos de Thierry Meyer Consultants

- Cabinet de consultants indépendants
- Créé en 2005 à Bordeaux

- Nos clients :
  - Entreprises, de la TPE au Grand compte
  - Institutions, administrations.

- Nos activités :
  - Ingénierie informatique
  - Sécurité informatique

## Pourquoi sécuriser mon site ?

- ”mon site c'est un petit site ...”
- Pourquoi me préoccuper de la sécurité ?
- En quoi ça consiste ?
- Pourquoi voudrait-on lui nuire ?
- Bon d'accord, mais comment on s'y prends ?

thierry.meyer@tm-consultants.fr

# La sécurité des applications Web

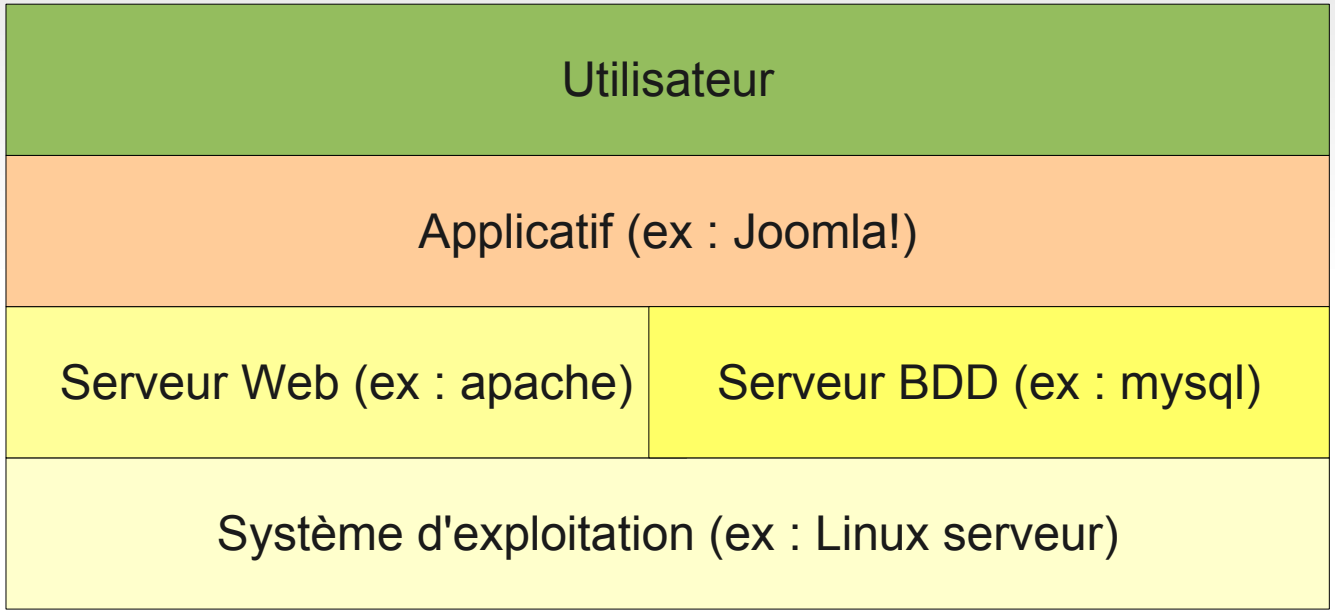
## La sécurité des applications Web

- Qu'est ce qu'une application Web ?
- Architecture et composantes
- A quel niveau se situe la sécurité ?
- Quelles sont les vulnérabilités des applications ?
- A propos des injections

# Qu'est ce qu'une application Web ?

- Application de type client/serveur
- Tourne dans un navigateur Web
- Utilise les technologies Web (HTTP, HTML/CSS ...)
- Fonctions avancées avec des langages dynamiques :
    - Javascript côté client
    - PHP côté serveur

# Architecture d'une application Web simple



## A quel niveau se situe la sécurité ?

- On compare souvent la sécurité avec une chaîne...
- Donc sécuriser **toutes** les couches de l'architecture
- Il n'y a pas de sécurité absolue !

# Quelles sont les vulnérabilités des applications ?

- Peut être liée à la technologie employée pour coder (*PHP register_global, ...*)
- Peut être liée à l'application elle même :
  - Architecture fragile
  - Installation / configuration
  - Fonctionnalités de l'applications
- Peut être des vulnérabilités communes aux applications Web en général (*injections sql, vol d'identifiant de sessions, ...*)

## A propos des injections

- Principal groupe d'attaques pour applications Web
- Type d'attaque générique visant à utiliser une interface d'entrée de l'application pour injecter du code ou détourner le fonctionnement du code existant.

# A propos des injections (2)

- Interface de communication avec l'application ?
  - Champ de saisie utilisateur
  - Url
  - Fichier de l'application corruptible
  - ...

# A propos des injections (3)

- Objectifs ?
  - produire un comportement inattendu de l'application (XSS,...)
  - obtenir des informations (injections SQL, CSRF, …)
  - corrompre les informations de la base (injections SQL)

# A propos des injections (4)

- Comment se protéger ?
  - Principalement former les développeurs à la sécurité
  - Acquérir les bons réflexes du premier coup
    - Validation des points d'entrée de l'application (*saisies utilisateurs,...*)
    - Choix d'une technique de programmation plutôt qu'une autre
    - Être auto-critique
    - ...

# Joomla! et la sécurité

thierry.meyer@tm-consultants.fr

# Joomla! et la sécurité

- Joomla est il (plus) sûr (qu'un autre) ?
- Les failles de Joomla!
- Renforcer la sécurité

## Joomla est il (plus) sûr (qu'un autre) ?

- Pas de réponse ferme sur le ”plus sûr”
- Possède des atouts et des faiblesses
- Peut-on malgré tout dormir tranquille ?
  (*oui à condition... :)* )

## Les atouts sécurité de Joomla!

- Architecture ”propre” => MVC
- Installation avec recommandations de sécurité
- Joomla! Security Strike Team (http://developer.joomla.org/security)

## Joomla! Security Strike Team

http://developer.joomla.org/security


- Equipe de 7 développeurs spécialisés.
- Rubrique du site Joomla!.org Dédiée sécurité
  - ”Joomla Security Center”
  - Conseils pratiques
  - Joomla! Administrators Security Checklist
- Mais /!\ en Anglais :(

## Les faiblesses de Joomla!

- Très grand choix d'extensions
  - Très pratique
  - Mais multiplie les risques d'autant
  - Toutes les extensions n'ont pas de suivie de mise à jour !!!

# Les failles de Joomla!

- Quelques exemples pratiques :
  - Joomla! <1.5.6 injections SQL
    + infos
    http://www.paperblog.fr/994900/joomlaune-faille-de-securite-importante-a-ete-detectee/
    (*démo*)
  - Capture du mot de passe Administrateur via Wireshark
    (*démo*)

# Renforcer la sécurité de Joomla!

- Inclure la sécurité comme contrainte du cycle de développement
  (SDLC Software Developpment Life Cycle )
- Utiliser un environnement de production sain
- Tenir son site à jour
- Mettre en oeuvre une politique de sauvegardes sùr

## Inclure la sécurité dans le SDLC

- Intéressant, travaux de l'OWASP *(http://www.owasp.org)*
- Mettre en oeuvre une architecture de développement
- Utiliser au moins 2 serveurs (idéalement 3 ) :
  - dev
  - (Pré-production)
  - Production
- Si mode ASP, bien sélectionner son hébergeur
  - Dédié ?
  - Mutualisé ?

## Inclure la sécurité dans le SDLC (2)

- Bien choisir ses extensions
- Former les développeurs (et chefs de projets) à la sécurité
- Etre vigilant si développement d'extensions
- Bien respecter les procédures ”d'override” si modification d'une extension.

## Utiliser un environnement de production sain

- Préparer son environnement
  - Configuration fine de PHP
  - Configuration adaptée d'Apache
  - Configuration du système de fichiers
  - Suppression des fichiers inutiles
  - ...

- Configuration de Joomla!
  - Renommer le compte administrateur (**+ mdp fort**)
  - Protéger la zone d'administration par HTTPs
  - Vérifier la liste des modules créés
  - Désinstaller tous les composants désactivés
  - Connaitre la liste des utilisateurs
  - ...

## Tenir son site à jour

- C'est une règle primordiale !
- Se tenir informé des vulnérabilités recensées (http://developer.joomla.org/security/news.html)
- Appliquer régulièrement les mises à jours (Joomla! **et** extensions !!)
  (prendre soin de sauvegarder avant !!)

## Mettre en oeuvre une politique de sauvegardes sùr

- Soit directement avec des outils de l'architecture (outils systèmes, mysql, ...)
- Soit à l'aide d'extensions (Joomlapack, ...)

## Voilà c'est (presque) tout :)

- En respectant ces quelques règles ”d'hygiène” cela devrait bien se passer.

  Merci.

# Questions ?