Sécurisation des applications web : retour d'expérience des développeurs de Sympa

David Verdin

-

TutoJ RES 12 : Sécurité des sites web

-

4 février 2010

En guise d'intro

- Le point de vue des développeurs.
  - Proposer des fonctionnalités
  - Permettre l'accès aux ressources
  - Rendre l'application conviviale et ergonomique

=>Rien à voir avec la sécurité... Pourtant la sécurité est une préoccupation constante, au même titre que la maintenabilité du code

  - Comment les besoins de sécurités se manifestent-ils ?
  - Comment avons-nous sécurisé progressivement Sympa ?
  - Quelles leçons tirons-nous de notre expérience ?

Le chemin suivi

- Sympa et ses vulnérabilités
- La sécurisation des mots de passe
- La sécurisation des sessions
- L'élimination des failles XSS
- Audit de sécurité
- Leçons tirées en terme de développement

Sympa

- Utilisé dans de très nombreuses organisations :
  - .gouv.fr : education, justice, recherche, diplomatie, culture, etc.
  - NASA, UNESCO, PCF, CGT, etc.

=> Sympa est s*usceptible* d'être attaqué

- Middleware : interagit avec de nombreuses applications

=> Les développeurs ne peuvent faire aucune hypothèses sur l'environnement dans lequel Sympa est mis en œuvre

Les points sensibles de Sympa

- Sa base de données :
  - adresses email
  - mots de passe
  - sessions
- Le fichier de config :
  - Identifiants base de données
  - Clés privées des certificats

Si le serveur est compromis, pas de salut pour Sympa

- Serveur dédié
- Serveur administré par des personnes de confiance

Les menaces

- Vol d'adresses email
  - 280 000 sur le serveur du CRU
- Envoi de messages :
  - outrepassant les règles d'une liste
  - Réputation du service : une fois suffit pour faire du mal...
- Création de liste :
  - Envoi de message libre
  - Utilisation du site de la liste à des fins illicites
- Confidentialité :
  - Archives
  - Documents partagés

La protection des mots de passe

Version de Sympa	Protection des mots de passe	Commentaires
Aube des temps	Stockage en clair	Pas de protection spécifique. Le bon vieux temps...
3.1 (2001)	Chiffrement symétrique	•Permet le rappel de mot de passe •si on perd le secret, on perd les mots de passe
5.4 (2008)	Chiffrement asymétrique	•Stock un hash MD5 •Plus de secret •Plus de rappel des mots de passe.
6.1 (2010)	Protection force brute	•Désactivation après 20 essais infructueux •Pas de test de Turing (SOAP)

Sessions dans Sympa

Version de Sympa	Gestion des sessions	Commentaires
Depuis toujours	Cookie = email + f(email,secret) => non forgeable Id = email	Intercepté => rejouable
5.4 (2008)	Id = Cookie = random()	Renouvelé à chaque clic => non rejouable Inconvénient : perte de session => pas de renouvellement en HTTPS
6.0 (2009)	Flag httponly positionné dans les entêtes HTTP	Javascript ne peut plus manipuler le cookie.

Les URL authentifiantes

- Exemple : message à modérer :
  - Un message avec une URL authentifiante est envoyé au modérateur
  - Le modérateur clique sur l'URL et arrive authentifié sur l'interface web de Sympa
- Fonctionnement : un ticket dans l'URL, le même dans la base de données
- Risque : forwarder un tel message
  - Les tickets expirent
  - Ils ne sont utilisables qu'une fois
  - Pas d'URL authentifiante pour se désabonner en bas de message

XSS

- Avant la version 5.4, Sympa ne filtrait pas les balises HTML

  => *XSS possible*

- Le problème : la diversité des injections possibles

  =>Voir : http://www.cru.fr/documentation/analyses/xss

  => *filtrage en entrée illusoire*

- Solution : filtrer en sortie

  => *tout paramètre affiché par Sympa est filtré : HTML supprimé*

  => *pour certains paramètres où le HTML est autorisé : utilisation d'une expertise externe, le module HTML::StripScripts*

Sympa est porteur sain.

Audit

- Conduit par P. Gardenat, Ac. Rennes : attaques sur Universalistes (pas seulement Sympa)
  - Une semaine de tests
- Résultats :
  - Restes de XSS : nous devons afficher du HTML issu de tierces parties (répertoires partagés et archives)
  - Des possibilités de CSRF
  - Attaque de force brute
- C'est corrigeable, mais
  - sécuriser l'interface SOAP (adresse IP)
  - Le SSO permet de « mutualiser » les XSS : on dépend aussi des autres applications.

Leçons

- Ne pas compter sur les autres, même pas les modules installés localement.
- Corriger les failles de sécurité ne suffit pas : encore faut-il que les utilisateurs mettent à jour leur installation !
  - Faciliter mises à jour
  - Faciliter packaging
- Règles de développement : tout ce qui entre et qui sort passe par des fourches caudines
  - Paramètres
  - Pages HTML générées par d'autres applications.

Conclusion

- Les gros bonds de sécurité ont eu lieu à la version 5.4 => mettez à jour
- Le travail continue :
  - À notre initiative pour les gros changements
  - À l'initiative d'utilisateurs qui nous rapportent des bugs.

Merci de votre attention