M1 - Outils de l'Internet (Année 2010/2011)

Mon nom est personne

L'objet de ce troisième TD est de comprendre le rôle et le fonctionnement des *cookies*, comment ils peuvent être utilisés pour implémenter des *sessions*, et comment on résout simplement (mais pas définitivement) le problème de l'identification (*i.e. authentification*).

Exercice 1. *Parlons de biscuits*

Le protocole HTTP est en protocole *state-less*, sans état. Bien que reposant sur TCP, les requêtes HTTP elles-mêmes sont semblables à un échange de deux datagrammes : on envoit une requête, on reçoit une réponse, fin de l'échange. Le protocole ne propose aucun mécanisme pour faire référence à des requêtes précédentes ; il n'a pas de mémoire, il n'y a pas de notion d'état. Les fonctionnalité de type `Keep-Alive` rajouté dans la version 1.1 du protocole permettent de réutiliser une même socket pour plusieurs requêtes mais ne changent rien au principe du protocole (de plus ce n'est qu'une recommandation que les serveurs peuvent choisir de ne jamais respecter). Cependant, dès qu'on construit un site dynamique un tant soit peu complexe, par exemple sur lequel des utilisateurs ont la possibilité de s'identifier, on ressent le besoin de simuler via HTTP des protocoles avec connexion. En un mot, de gérer des *sessions* ouvertes par des *utilisateurs*.
La mise en œuvre des sessions ne peut se faire sans la collaboration (volontaire ou non) du navigateur web : une identification reposant sur l'adresse IP du client ne fonctionnerait pas puisque plusieurs clients pourraient être en train de se connecter à partir de la même adresse IP en même temps. De plus, un même client pourrait vouloir gérer plusieurs sessions avec le même serveur en parallèle.

1. Proposez un mécanisme simple utilisant le passage de paramètres (par la méthode GET ou POST) pour associer un état à une requête web. Discutez des avantages de l'une et l'autre méthode.

2. Quels problèmes pose ce type de mécanisme vis-à-vis : de la confidentialité des données stockées par un proxy, de la persistance (que se passe-t-il si l'utilisateur ferme son navigateur par erreur et revient ensuite sur le site ?)

3. Un mécanisme plus général mis en place au niveau des en-têtes HTTP pour résoudre le problème de la gestion de session et de la persistance des données entre connexions est celui des cookies. Ce mécanisme a été introduit par Netscape (qui fait référence), puis formalisé dans la RFC2109 (que personne ne respecte) et étendu dans la RFC2965. Il permet au serveur de demander au client de stocker des valeurs de variables et de les lui renvoyer dans les requêtes suivantes, sans que la fermeture du navigateur ait une incidence.

4. Commencez par lire les deux documents joints : la documentation Netscape sur les cookies et l'extrait du fichier de cookies appartenant à un utilisateur anonyme.

5. Proposez une utilisation simple des cookies pour gérer les préférences d'affichage d'un utilisateur (langue, habillage des pages), un panier d'achats et un mécanisme de login sur un même site web. Donner quelques échanges typiques d'entêtes `Cookie` et `Set-Cookie` lors de la consultation du site.

6. Expliquez les mécanismes à mettre en oeuvre pour implémenter les cookies : du côté du serveur et du côté du navigateur.

7. Lisez les cookies fournis. Quels types d'utilisation y distinguez-vous ? La date d'expiration est donnée en nombre de secondes depuis l'Epoch (aujourd'hui midi 1255345200, dans un an 1286881200, dans cinq ans 1413111600). Que pouvez-vous dire des dates d'expirations utilisées par les sites ?

8. Les cookies peuvent être détournés dans des buts marketing. Identifiez deux façons différentes de le faire. Expliquez comment la norme protège les utilisateurs de certaines utilisations déviantes... mais pas de toutes !

Exercice 2. *Parlons de sessions*

Un mécanisme de gestion de session est généralement mis en oeuvre en utilisant des cookies pour stocker tout ou partie de l'état de la session courante de l'utilisateur. Dans le cas où le client ne permet pas l'utilisation des cookies (par incapacité ou par choix politique), une bonne politique est de dégrader le service vers l'utilisation d'un mécanisme à base de variables POST ou GET (comme décrit en début de TD). Discutons de la bonne manière de gérer les variables de session.

Utilisons le scenario suivant. Notre site web propose des quizz. L'utilisateur a la possibilité de paramétrer la couleur de fond des pages du site. Lorsqu'il se connecte, il choisit un quizz puis le parcourt question par question. À la fin, on affiche son pourcentage de réponses correctes. En pratique, le site a besoin de mémoriser le numéro de la question courante et le nombre de réponses correctes données jusque là.

1. Quelles données faut-il stocker dans les cookies ? Comment tricher si toutes les données sont du côté client ?

2. On décide de stocker un numéro de session dans les cookies pour les données « sensibles ». Que faut-il mettre en oeuvre côté serveur pour utiliser un tel mécanisme ?

3. Identifiez des problèmes de sécurité : peut-on usurper l'identité de quelqu'un en écoutant le flux réseau ? peut-on rejouer plusieurs fois en stockant et réutilisant les cookies ? peut-on deviner facilement des numéros de session valides ? Proposez des mécanismes pour améliorer la gestion des sessions (sur les points où c'est possible).