Sécurité en ingénierie du Logiciel

Le cadre des Web Services
Partie 5 : Authentification

Alexandre Dulaunoy

adulau@foo.be

Agenda

- Introduction
- Authentification HTTP
- HTTP Basic
- HTTP Digest
- Authentification X.509/SSL
- Authentification Services Web/applicative
- Conclusion
- Exemples

Introduction

Authentification et Services Web
L'authentification utilisateur(*) est le processus pour vérifier l'authenticité d'un utilisateur(*). Utilisateur peut être une entité.

- Plusieurs approches (difficile de choisir),
- Questions de compatibilité entre browser/serveur,
- Persistances de l'authentification,
- Sécurité de l'authentification,

Authentification HTTP

- HTTP est "stateless" (cf. cours 2),
- Authentification HTTP/1.1 -> RFC2617,
  - Basic et Digest Access Authentification
- Valide pour les Web Services utilisant HTTP,
- mais aussi pour toutes communications HTTP,

HTTP Basic

L'authentification "Basic" est basée sur le principe le client doit s'authentifier pour chaque "realm". Le "realm" est une valeur opaque.
realm="WallyWorld"
La chaine d'authentification est au format :
Authorization: Basic <base64 user:password>
p.ex :
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==

(encodée en Base64)

HTTP Basic : exemple

- Le client émet une requête :
  GET /admin/ HTTP/1.1
- Le serveur répond avec une entête (header) WWW-Authenticate :
  HTTP/1.1 401 Authorization Required
  WWW-Authenticate: Basic realm="Admin Password"
- Le client (suite à la saisie d'un password) émet une nouvelle requête avec :
  GET /admin/ HTTP/1.1
  Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==

HTTP Digest

Pour résoudre plusieurs problèmes de HTTP Basic :

- L'interception du password,
- Le rejeu possible (replay attack),
- La vérification du client ET du serveur,
- Le stockage des password sur le serveur,
- La vérification de l'URI,...

HTTP Digest

- L'authentification "Digest" est basée sur le principe "challenge/response". Le serveur envoie un challenge au client pour chaque "realm".
- Une fonction de "hashage" est utilisée dans le cadre du "challenge/response" pour ne pas divulguer le secret partagé.

Apache (mod_auth_digest).

HTTP Digest : Calcul

request-digest = <"> < KD (H(A1), unq(nonce-value)
":" nc-value
":" unq(cnonce-value)
":" unq(qop-value)
":" H(A2)
) <">
KD (secret,data) = H(concat(secret, ":", data)
A1 = unq(username-value) ":" unq(realm-value) ":" passwd
A2 = Method ":" digest-uri-value

HTTP Digest : Exemple

- Le client émet une requête :
  GET /admin/ HTTP/1.1
- Le serveur répond avec une entête (header)
  WWW-Authenticate:

HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
realm="testrealm@host.com",
qop="auth",
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
opaque="5ccc069c403ebaf9f0171e9517f40e41"

HTTP Digest : Exemple

- Le client répond (après le calcul de response) :

Authorization: Digest username="Mufasa",
realm="testrealm@host.com",
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
uri="/admin/",
qop=auth,
nc=00000001,
cnonce="0a4f113b",
response="6629fae49393a05397450978507c4ef1",
opaque="5ccc069c403ebaf9f0171e9517f40e41"
Le serveur répond aussi une "reponse" (calcul identique que pour le client sauf pour A2 (:/admin/ uniquement)).

HTTP auth - suivi authentification

- HTTP Basic
  - Transmission à chaque requête de l’entête.
- HTTP Digest
  - Transmission à chaque requête de l’entête (et donc recalcul).

Authentification X.509/SSL

- Authentification dans le protocole TLS/SSL,
- Authentification X.509 unilatérale (le serveur uniquement),
- Authentification X.509 mutuelle (serveur et client),
- Comme TLS/SSL encapsule HTTP, une authentification "Basic" peut être utilisée sans divulguer le password

Authentification X.509/SSL - suivi de session

- Par session TLS/SSL, il y a une référence (SSL_SESSION_ID) qui permet de suivre une session authentifiée,
- Export des paramètres X.509 (DistinguishedName, CommonName...),
- Durée de vie des sessions (24 heures suivant RFC TLS),
- Session authentifiée au niveau "applicatif".

Authentification Applicative / Services Web

Comment maintenir une authentification de session applicative ?

- Cookies et Secure Cookies (TLS),
- Method GET/POST avec id de session,
- Header de session propre au serveur,
- Insertion dans les URI,
- ....

Q&R

- adulau@foo.be
- http://www.foo.be/cours/securite-webservices/
- 3B12 DCC2 82FA 2931 2F5B 709A 09E2 CD49 44E6 CBCD