TLS/SRP(RFC5054), une histoire sans fin avec OpenSSL (et d'autres) ?
-
Le projet EdelKey

Peter.sylvester@edelweb.fr

Réunion OSSIR 11 octobre 2011

SRP dans son écosystème

- password enhanced keyexchange and authentication
- à la mode il y a 10 ans
- Les trolls des brevets
  - Stanford, Lucent, Phoenix
- IEEE-P1363 (un peu usine à gaz)
- IETF (TLS, PKIX, SACRED, ...)

Les textes IETF

- T. Wu, Stanford University : RFC 2945
  - Septembre 2000
  - The SRP Authentication and Key Exchange System
- RFC 5054
  - Novembre 2007, premier draft février 2001
  - Using the Secure Remote Password (SRP) Protocol for TLS Authentication

Nos débuts chez EdelWeb

- 2000 IETF Adelaide
  - Discussion du groupe de travail SACRED, steak de Kangaroo
- 2002: utilisation de certificats (la clé privée) avec roaming.
  - Idée: pas de téléchargement, mais l'accès à distance
- Les premiers stages 2003
  - PKCS11 et « rpc » sur TLS
  - Traceur PKCS11
  - Etude sur SRP
  - Implémentation PKCS11 sur PKCS12

Le projet / des sujets de stage

- Bibliothèque PKCS11
  - Communication TLS/SRP
  - Un protocole RPC pour C_Encrypt
- Un serveur apache modifié
  - + cgi qui s'appuie sur PKCS12
- Intégration avec curl.
- ...

SRP, comment ça marche

- Idée: modifier l'échange Diffie-Hellman
- Définitions
  - N, g: group parameters (prime and generator)
  - s: salt
  - B, b: server's public and private values
  - A, a: client's public and private values
  - I: user name (aka "identity")
  - P: password
  - k: SRP-6 multiplier
- Préparation:
  - x = SHA1(s | SHA1(I | ":" | P))
  - v = g^x % N

SRP/TLS, comment ça marche?

- Le protocole
  - Modifie l'échange DH
  - Dans TLS sans modification du handshake (SRP-6 et rfc 5054)
- Utilisation des messages
  - ClientHello (1)
  - ServerHello (2)
  - ServerKeyChange (3)
  - ClientKeyExchange (4)
- Finished message
  - Pour vérifier si on est bon,
  - établissement de clé partagé et authentification mutuelle

Calcul du premaster secret

- Calcul premaster secret par le client
  - I, P = <read from user>
  - N, g, s, B = <read from server>(3)
  - a = random()
  - A = g^a % N
  - u = SHA1(PAD(A) | PAD(B))
  - k = SHA1(N | PAD(g))
  - x = SHA1(s | SHA1(I | ":" | P))
  - <ps>= (B - (k * g^x)) ^ (a + (u * x)) % N

- Calcul premaster par le serveur
  - I <read from client>(1)
  - N, g, s, v = <read from password file>
  - b = random()
  - k = SHA1(N | PAD(g))
  - B = k*v + g^b % N
  - A = <read from client>(4)
  - u = SHA1(PAD(A) | PAD(B))
  - <ps> = (A * v^u) ^ b % N

SRP/TLS, la coordination

- Définition des ‘ciphersuites’
  - Et la bagarre commence avec IETF-TLS
    - changée entre les drafts et la version définitive
- Le numéro d’extension
  - a été réservé par l’IANA
    - mais donné à quelqu’un d’autre ensuite
- Un changement du protocole
  - Plus de protection d’une reprise

Implémentation OpenSSL

- Support des extension TLS
  - ServerNameIndication
  - Extension SRP/TLS
    - Approche avec #ifdef
- Un module crypto pour les calculs
- Définition de « ciphersuites »
- Modifications du handshake TLS
- Modifications s_client et s_server
- Outil de génération de vérificateur

Implémentation OpenSSL

- ServerNameIndication pour commencer
  - Extension simple, permet de se familiariser
  - Vérification de la compétence du stagiaire
- Résultat après 2-4 semaines?
- Cette partie a été isolée ensuite pour être intégrée.
  - En 2006 par Bodo Müller
  - Accident: en même temps qu'une autre extension
  - Conséquence: la logique des extensions est &é »'(-è.
  - En 2008, support en apache par Kaspar Brand
- Et SRP, on est où?

Et SRP/TLS ?

- Implémentation fini en 2004
  - Implémentation complet pendant le stage
  - Modification apache + cgi aussi
  - Modification PKCS11 aussi
- D'autres stages
  - Etude NSS, refaire apache, …

- Projet présenté à la EFPE 2005 (Pologne)
- Article dans un workshop PKI du DFN

- Tom WU embauché chez CISCO

La vie continue

- Tom WU a analysé le patch et fourni une nouvelle version pour plusieurs versions d'OpenSSL.
  - Pas de changement de logique
  - Sha1 par « evp »
- Enfin, en 2011, le patch à été intégré dans la version de dev.
  - 1% du code
  - GNU/TLS était en avance, bouncycastle aussi
  - support dans curl
  - En attendant apache, NSS, ...

SRP/TLS et la suite

- Les PAKE, c'est à la mode
  - Transmettre un code PIN (carte)
    - On peut se mettre en PIM électrique
- Authentification dans des clients email etc.
  - À la place d'authentification serveur + mdp en « clair »
    - DigiNotar et Comodo
- Intégration apache, stunnel, nodejs, NSS, Thunderbird
  - pas facile

La sécurité et les trolls

- Tom Wu pense que c'est équivalent à DH (il n'est pas seul).
- Pas d'attaque hors ligne.
- Il faut protéger les vérificateurs et la saisie du mot de passe.

- Des faux problèmes
  - « SRP ne marche pas avec des courbe « elliptique », on tourne en rond?
  - « Le I est en clair, je m'appelle PS »

- Les brevets commence à expirer (EKE)
  - Le brevet de Stanford est défensif

Résumé

- Mode cathédrale (OpenSSL, apache)
- IETF, les lobbies, les trolls, …
- Il y a des amis.
- Il ne faut pas abandonner.
- C’est drôle quand-même.

- Et la suite?

Les liens de la fin

- http://www.edelweb.fr/EdelKey/
  - Historique du projet (et le future?)
- http://srp.stanford.edu/
  - Le site de référence
- http://en.wikipedia.org/wiki/Secure_Remote_Password_protocol
  - Un point d'entrée