[Sécurisation des communications avec SSL v.3]

Février 2001

http://securit.free.fr

Sommaire

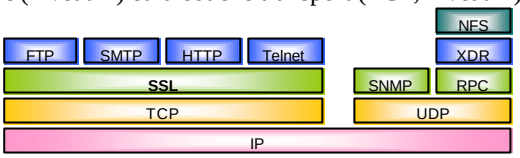
I. Présentation et architecture

Présentation

- **Historique :** SSL développé par Netscape
  - ✓ Première version testée en interne
  - ✓ Version 2.0 publiée en 1994
  - ✓ Version actuelle 3.0
    - *Draft IETF publié en 1996*
    - *En cours de standardisation à l'IETF au sein du groupe Transport Layer Security*
- **Objectifs de SSL v.3**
  - ✓ Fournir des services de sécurité basés sur des moyens cryptographiques :
    - *Authentification (unidirectionnelle ou bidirectionnelle)*
    - *Confidentialité des données*
    - *Intégrité des données*
  - ✓ Interopérabilité
  - ✓ Extensibilité et efficacité

Architecture

- Pile TCP/IP : SSL est une couche qui se situe entre les services applicatifs (niveau 7) et la couche transport (TCP, niveau 4)

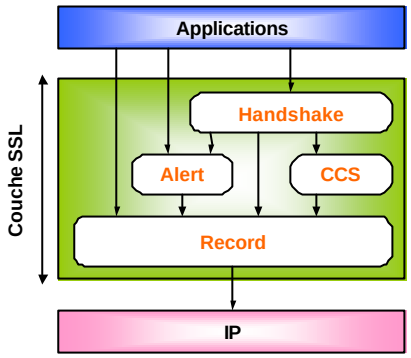


- SSL est composé de deux niveaux
  - ✓ 1er niveau : protocole au-dessus de TCP/IP
    - *Record Layer Protocol*
  - ✓ Niveau supérieur : 3 sous-protocoles
    - *Handshake Protocol*
    - *Change Cipher Spec Protocol*
    - *Alert Protocol*

II. Sous-protocoles

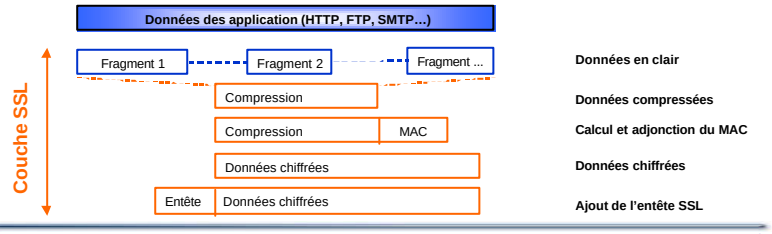
Sous-protocoles

- Les 4 sous-protocoles de SSL



Sous-protocole RECORD

- Fonction du sous-protocole RECORD
  - ✓ Réception des données des couches supérieures
  - ✓ Traitement des paquets de données :
    - *Fragmentation en blocs de taille maximum 2^{14} octets et compression*
    - *Calcul d'intégrité par génération d'un condensât*
    - *Chiffrement des données*
  - ✓ Transmission des données au protocole TCP

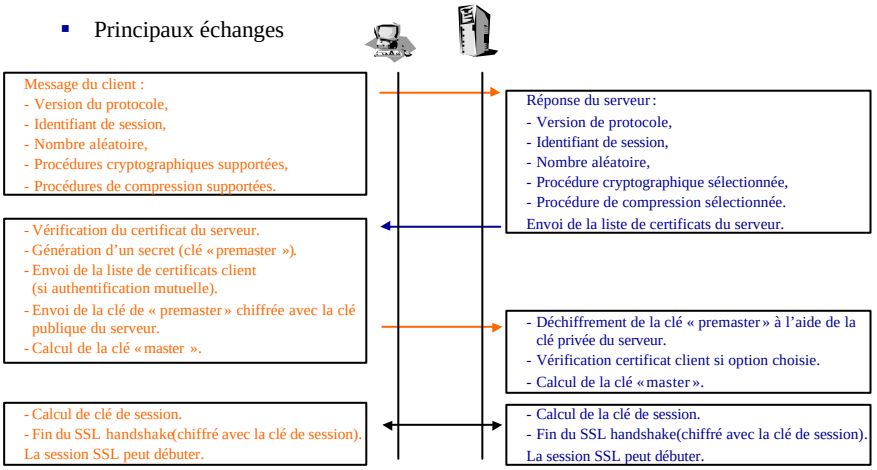


Sous-protocole HANDSHAKE

- Objectifs
  - ✓ Authentification du serveur
  - ✓ Authentification optionnelle du client
  - ✓ Négociation des algorithmes utilisés (chiffrement et MAC)
  - ✓ Génération de la clé symétrique de session pour le chiffrement des données
- Notions
  - ✓ Connexion
    - *Lien logique entre un client et un serveur. Une connexion est to ujours associée à une session.*
  - ✓ Session
    - *Association entre un client et un serveur, créée par le protocole Handshake. Une session définit les paramètres de sécurité qui peuvent être partagés entre des connexions multiples (évite de négocier des nouveaux paramètres de sécurité pour chaque connexion)*

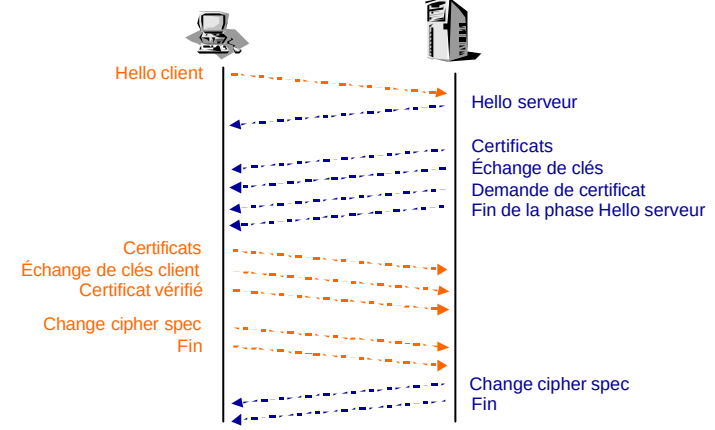
Sous-protocole HANDSHAKE

- Principaux échanges



Sous-protocole HANDSHAKE

- Cinématique de l'ouverture d'une session SSL v.3



Sous-protocole HANDSHAKE

- **Hello client**
  - ✓ Version de SSL
  - ✓ Nombre aléatoire : *client_random*
  - ✓ Numéro de session : *session_ID*
  - ✓ Liste des suites de chiffrement choisies par le client
  - ✓ Liste des méthodes de compression choisies par le client
- **Hello serveur**
  - ✓ Version de SSL
  - ✓ Nombre aléatoire : *server_random*
  - ✓ Numéro de session : *session_ID*
  - ✓ Une suite de chiffrement choisies par le client
  - ✓ Une méthode de compression choisies par le client
- **Algorithmes négociés par le protocole HANDSHAKE**
  - Échange de clés
    - ✓ *RSA, Diffie-Hellman, Fortezza*
  - Chiffrement symétrique à la volée
    - ✓ *RC4 (40 ou 128 bits)*
  - Chiffrement symétrique en bloc :
    - ✓ *DES, 3DES, RC2, IDEA, Fortezza*
  - Fonction de hachage
    - ✓ *MD5, SHA-1*

[Sécurisation des communications avec SSL v.3]

Sous-protocole HANDSHAKE

- Variables d'état d'une session SSL
  - ✓ Identificateur de session (*session_ID*)
  - ✓ Certificat du pair (*peer certificate*)
  - ✓ Méthode de compression (*compression method*)
  - ✓ Suite de chiffrement (*cipher spec*)
  - ✓ *MasterSecret* de 48 octets partagés entre le client et le serveur
  - ✓ Drapeau (*is resumable*)
  - ✓ Suite de chiffrement contenant les cinq éléments suivants :
    - *Mode de chiffrement utilisé*
    - *Algorithme de chiffrement utilisé*
    - *Algorithme de hachage utilisé*
    - *Taille du condensât*
    - *Permission d'exporter l'algorithme de chiffrement*

[Sécurisation des communications avec SSL v.3]

Sous-protocole HANDSHAKE

- Variables d'état d'une connexion SSL
  - ✓ Deux nombres aléatoires de 32 octets
    - *server random*
    - *client random*
  - ✓ Deux clés secrètes pour le calcul des codes d'authentification
    - *server write MAC secret (pour les données chiffrées par le serveur)*
    - *client write MAC secret (pour les données chiffrées par le client)*
  - ✓ Deux clés pour le chiffrement symétrique
    - *server write key*
    - *client write key*
  - ✓ Vecteurs d'initialisation pour le chiffrement symétrique en mode CBC
  - ✓ Numéros de séquence l'un pour le serveur et l'autre pour le client

Autres sous-protocoles

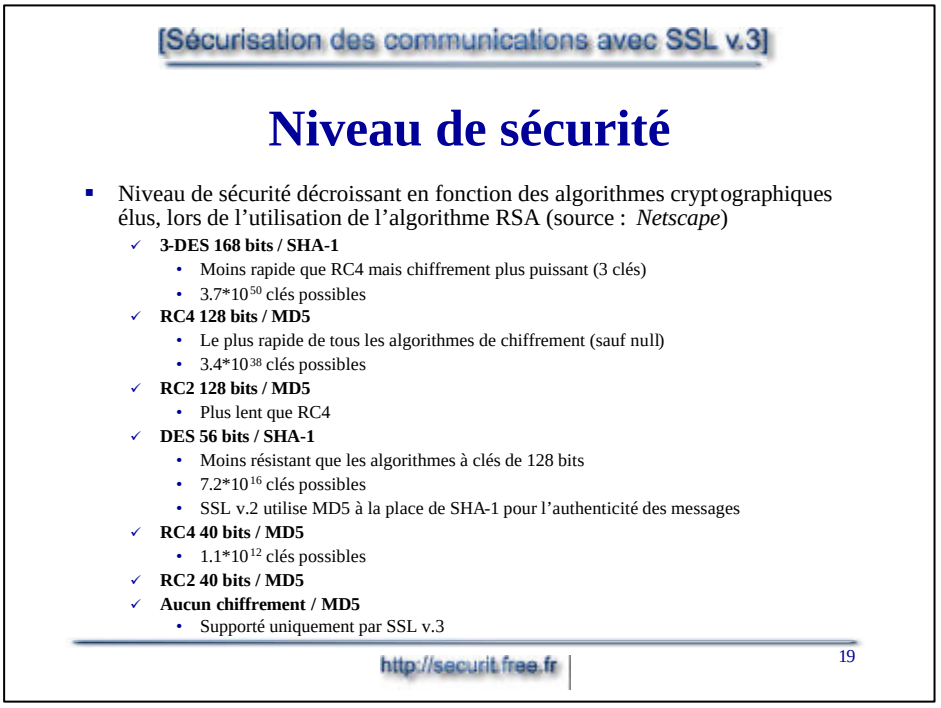
- Change Cipher Spec
  - ✓ Objectifs
    - *Signale au protocole RECORD toute modification des paramètres de sécurité*
  - ✓ Moyens
    - *Message bi-directionnel signale la transition de la stratégie cryptographique en cours (« pending state » devient « current state »)*
- Alert
  - ✓ Objectifs
    - *Signale les alertes*
    - *Alert peut être invoqué par :*
      - Une application, pour signaler par exemple la fin d'une connexion.
      - Le protocole Handshake suite à un problème survenu au cours de son déroulement.
      - Le protocole Record si, par exemple, l'intégrité d'un message est mise en doute.
  - ✓ Moyens
    - *2 octets contiennent le niveau de sévérité et la description de l'alerte*

III. SSL et les certificats

SSL et les certificats

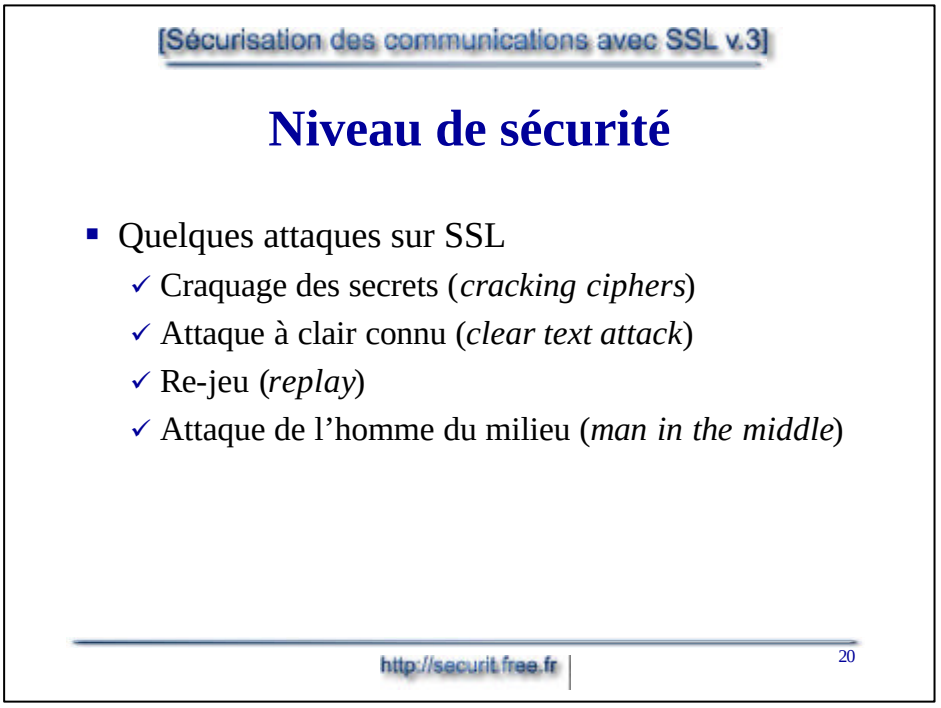
- SSL v.3 permet l'authentification mutuelle (client et serveur).
- Optionnellement, l'échange de certificat client-serveur peut être réalisé.
- Normes supportées : X.509 v.3, PKCS#
- Voir à ce sujet:
  - ✓ http://www.ultranet.com/~fhirsch/Papers/wwwj/article.html

IV. Analyse du niveau de sécurité



Source NETSCAPE :

http://developer.netscape.com/docs/manuals/security/sslin/index.htm



Craquage des secrets (cracking ciphers)

Cryptanalyse pour factoriser la clé publique d'un serveur et en découvrir la clé privée.

Écoute des échanges complets d'une session pour obtenir la clé de session.

Attaque à clair connu (clear text attack)

Attaques par dictionnaire sur un clair connu ou pressenti (connaissance des messages échangés).

Attaque largement réalisable sur SSL. Ex. l'un des clairs les plus connus : commande HTTP « GET ».

Solution : utiliser des des clés de session de taille suffisante et de bonne qualité !

Re-jeu (replay)

Ré-émission d'un échange autorisé et enregistrée par un pirate.

Parade SSL : mise en œuvre de numéros de séquence (« *nonce* » : *connection_id* du message *SERVER_HELLO*) calculés sur la base d'évènements aléatoires non prédictibles, de taille 128 bits.

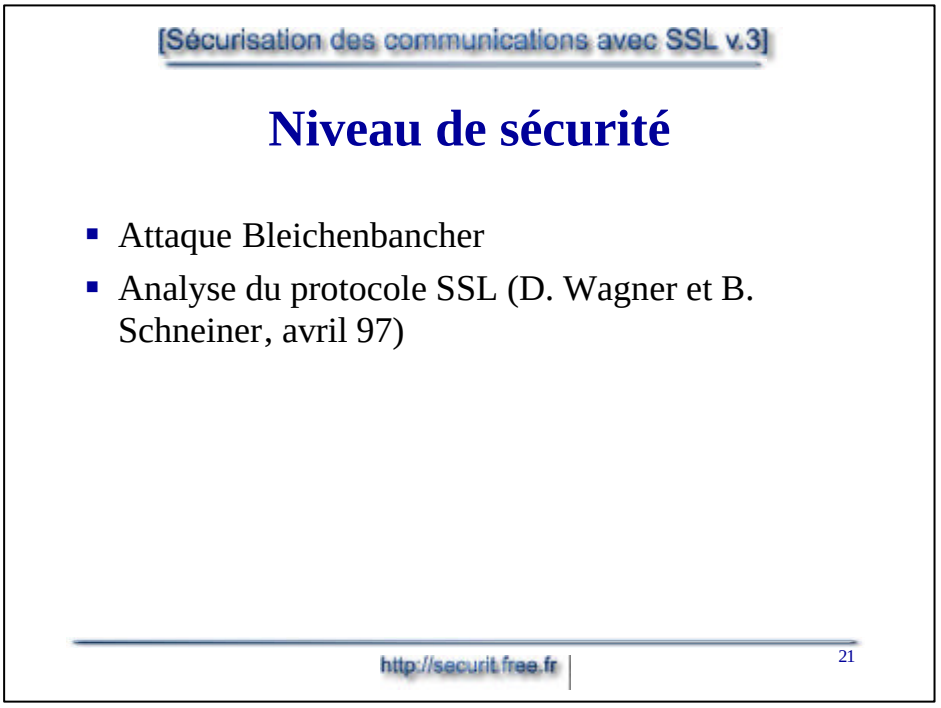
Attaque dite de l'homme du milieu (man in the middle)

Un pirate s'interpose entre le client et le serveur et tente d'usurper l'identité du serveur vis-à-vis du client (cas de Dsniff).

Solutions :

-Toujours utiliser des certificats serveur signés par des autorités de certification de confiance.

-Éduquer les utilisateurs à toujours vérifier le contenu d'un certificat (mapping entre le nom du serveur et le certificat).



Attaque Bleichenbancher

Une session SSL peut être déchiffrée par une analyse mathématique complexe et des envois de message d'essais en erreur :

- L'attaque nécessite environ un million de messages et n'est valable que pour une transaction. Elle ne peux pas être divisée en plusieurs machines pour réduire le temps. L'attaque doit être concentrée au niveau d'un seul serveur.

La parade à cette attaque doit se faire au niveau du serveur. Par exemple Microsoft a modifié une DLL d'IIS pour répondre à cette vulnérabilité.

Attention : permettre la compatibilité avec SSL v.2 peut être dangereux ! (failles de sécurité importantes)

Analyse du protocole SSL (D. Wagner et B. Schneiner, avril 97)

Record protocol

SSL dispose d'une protection correcte contre les attaques sur la confidentialité (écoute, analyse du traffic), l'authentification (MAC) et le re-jeu.

Handshake protocol

Présente des vulnérabilités graves liées à la négociation des clés :

Effacement du message change cipher spec : le message peut être enlevé et si l'implémentation ne l'interdit pas, la sécurité négociée n'est pas activée.

Réduction du niveau de sécurité de l'algorithme d'échange de clés : peut aboutir à la découverte de la clé maître (peut être corrigé par implémentation).

Niveau de sécurité

- La vérification des certificats, un enjeu de sécurité dans SSL :
  - ✓ Fonctionnement de Dsniff
    - *www.monkey.org/~dugsong/dsniff*
  - ✓ Le problème de la révocation des certificats
    - *http://www.cert.org/advisories/CA-2000-19.html*
  - ✓ Les certificats et la résolution des noms DNS
    - *http://www.cert.org/advisories/CA-2000-08.html*

V. Transport Layer Security

TLS v1.0

- Qu'est-ce-que TLS ?
  - ✓ Travaux de standardisation de SSL au sein de l'IETF, groupe de travail TLS
  - ✓ RFC 2246
- Comparaison SSL/TLS
  - ✓ TLS 1.0 est similaire à SSL v.3 (TLS ~ SSL v.3.1)
  - ✓ Différences faibles mais **TLS et SSL v.3 ne sont pas inter-opérables :**
    - *Algorithmes cryptographiques (calcul du MAC pour les clés de session)*
    - *Codes d'alertes (TLS définit de nouveaux codes d'alerte)*
    - *TLS spécifie cependant un mécanisme pour permettre une compatibilité ascendante avec SSL*

VI. Conclusions

Conclusions

- SSL v.3 est le moyen d'implémenter des VPN le plus utilisé au monde (VPN de niveau applicatif).
- SSL v.3 bénéficie de longues expériences et d'une maturité certaine.
- Les problématiques actuelles de PKI intègrent majoritairement SSL, mais des évolutions sont nécessaires pour fournir tous les services de sécurité.

VII. Annexes & Références

Références

- Liens
  - ✓ Informations Netscape
    - *http://home.netscape.com/eng/ssl3/ssl-toc.html*
    - *http://developer.netscape.com/docs/manuals/security/sslin/index.htm*
  - ✓ Analysis of the SSL v.3 protocol – D. Wagner/B. Schneier, April 1997
    - *http://www.counterpane.com/ssl-revised.pdf*

Annexe A

Ports des applications utilisant SSL

- Ports TCP/IP attribués par l'IANA

Protocole	Port	Description	Application
HTTPS	443	HTTP sécurisé	Transactions requête-réponse
SSMTP	465	SMTP sécurisé	Messagerie
SSNNTP	563	NNTP sécurisé	News réticulaires
SSL-LDAP	636	LDAP sécurisé	Annuaire
SPOP3	995	POP3 sécurisé	Accès distant à la messagerie

- Ports TCP/IP utilisés sans attribution formelle

Protocole	Port	Description	Application
FTP-DATA	889	FTP sécurisé	Transfert de fichiers
FTPS	990	FTP sécurisé	Contrôle de transfert de fichiers
IMAPS	991	IMAP4 sécurisé	Accès distant à la messagerie
TELNETS	992	Telnet sécurisé	Emulation de terminal
IRCS	993	IRC sécurisé	Protocole de conférence réticulaire ("chat")

Annexe B

Messages SSL

Message	Type de message	Sens de transmission	Signification
HelloRequest	Optionnel	Serveur -> Client	Notification au client pour débuter la phase de Handshake
ClientHello	Obligatoire	Client -> Serveur	Message contenant : - Numéro de version SSL - Nombre aléatoire :*client_random* - Numéro de session : *session_ID* - Suites de chiffrement choisies par le client - Méthodes de compression choisies par le client
ServerHello	Obligatoire	Serveur -> Client	Message contenant : - Numéro de version SSL - Nombre aléatoire : server_random - Numéro de session : session_ID - Une suite de chiffrement - Une méthode de compression
Certificate	Optionnel	Serveur -> Client Client -> Serveur	Message contenant : - Le certificat du serveur - Le certificat du client si le serveur lui a réclamé et si le client en possède un
ServerKeyExchange	Optionnel	Serveur -> Client	Message envoyé par le serveur si : - Il ne possède pas de certificat - Il ne possède qu'un certificat de signature
CertificateRequest	Optionnel	Serveur -> Client	Message envoyé par le serveur pour réclamer un certificat au client
ServerHelloDone	Obligatoire	Serveur -> Client	Message signifiant au client la fin des échanges liés à la phase *ServerHello* et subséquents
ClientKeyExchange	Obligatoire	Client -> Serveur	Message contenant le PreMasterSecret chiffré à l'aide de la clé publique du serveur
CertificateVerify	Optionnel	Client -> Serveur	Message permettant une vérification explicite du certificat client
Finished	Obligatoire	Serveur -> Client Client -> Serveur	Message signifiant la fin du protocole Handshake et le début de l'émission des données, protégées avec les nouveaux paramètres négociés

Annexe C

Messages du protocole ALERT

Message	Contexte	Type
Bad_certificate	échec de vérification d'un certificat	fatal
Bad_record_mac	réception d'un MAC erroné	fatal
Certificate_expired	certificat périmé	fatal
Certificate_revoked	certificat mis en opposition	fatal
Certificate_unknown	certificat invalide pour motifs autres que ceux précisés précédemment	fata
Close_notify	interruption volontaire de session .	fatal
Decompression_failure	la fonction de décompression ne peut pas s'appliquer (cas de données trop longues)	fatal
Fandshake_failure	impossibilité de négocier des paramètres satisfaisants	fatal
Illegal_parameter	la taille du paramètre échangé au cours du protocole Handshake dépasse la taille du champ correspondant ou lorsqu'il est sans rapport avec les autres paramètres	fatal
No_certificate	réponse négative à une requête de certificat	avertissement ou fatal.
Unexpected_message	arrivée inopportune d'un message	fatal
Unsupported_certificate	certificat reçu n'est pas reconnu par le destinataire	avertissement ou fatal