Services de la sécurité

- Confidentialité
- Intégrité
- Authentification

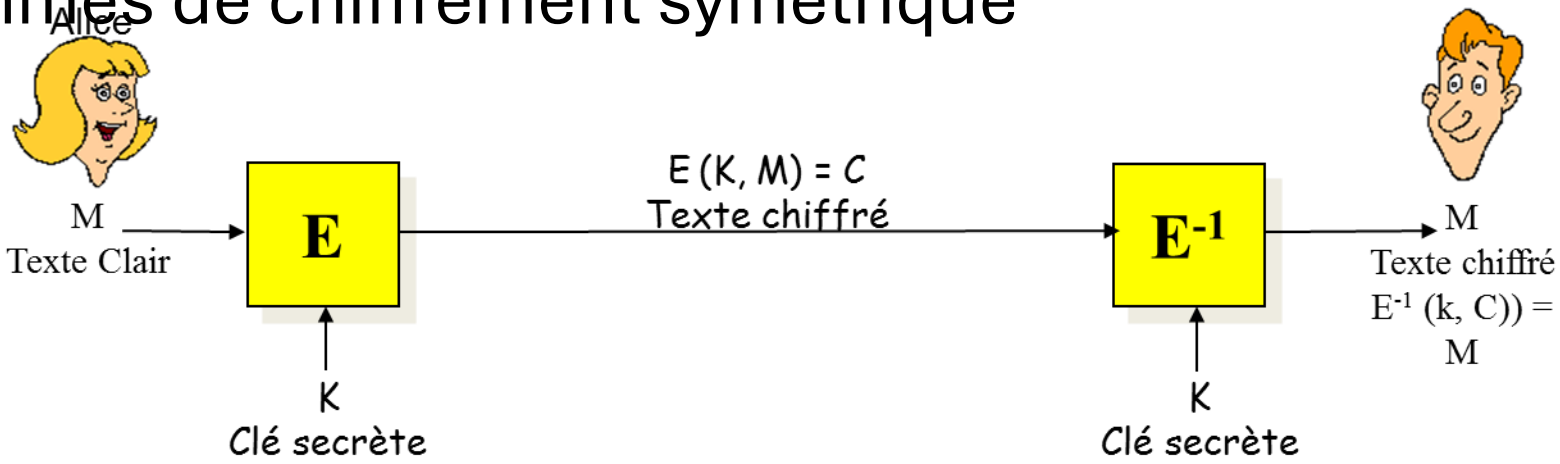
- **Identification**
- **Non répudiation**
- **Horodatage**

Eléments de cryptographie

- Algorithmes de chiffrement symétrique
  - Fonction de transformation basée sur deux paramètres
    - Une clé = k
    - Un message = M
  - Soit $E(k, M) = C$
  - Soit $E^{-1}(k, C) = M$
  - E et E^{-1} peuvent être identique
  - Exemple le ou exclusif

Eléments de cryptographie

- Algorithmes de chiffrement symétrique



- Problème d'échange de clé
  - Le réseau n'est pas de confiance
- Une clé par couple
  - Gestion et renouvellement

Eléments de cryptographie

- Algorithmes de chiffrement symétrique

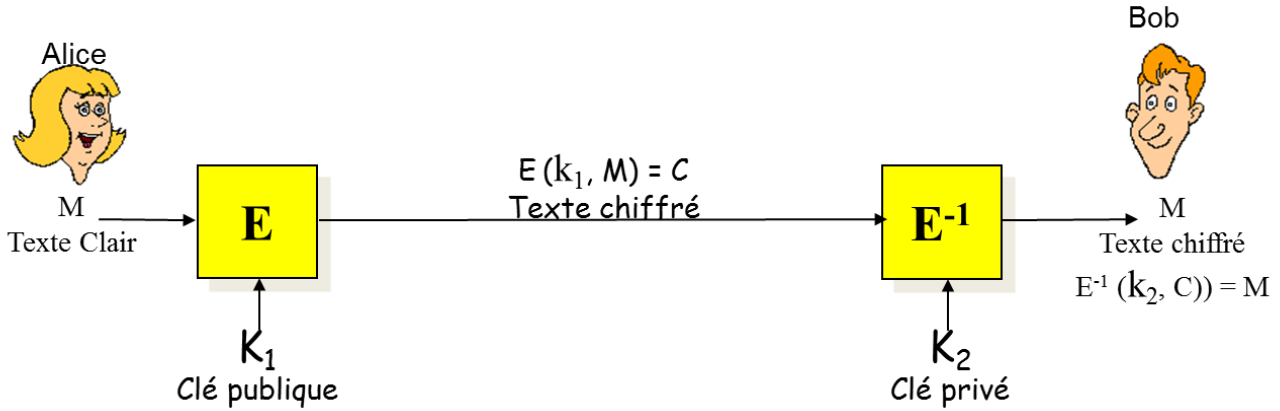
Algorithme	Nom et commentaires	Type de chiffrement	Longueur de la clé en bits	Normalisé
DES	*Data Encryption Standard*	bloc de 64 bits	56	FIPS Pub 81,1981 ANSI X3.92, X3.105, X3.106 ISO 8372 ISO/IEC 10116
IDEA	*International Data Encryption Algorithm,*	bloc de 64 bits	128	
RC2	développé par Ronald Rivest	bloc de 64 bits	variable, 40.exp.	Non et propriétaire
RC4	développé par R. Rivest	enfilé	variable 40 - 128	Non, diffusé sur l'Internet en 1994
RC5	développé par R. Rivest	bloc de 32, 64 ou 128 bits	variable à 2048	Non et propriétaire
SKIPJACK	Confidentiel NSA .	bloc de 64 bits	80	Secret défense US
Triple DES		bloc de 64 bits	112	ANSI X9.52
AES	Advanced Encryption Standard	bloc de 128 bits	128,192, 256	FIPS197 2001

Eléments de cryptographie

- Algorithmes de chiffrement symétrique
  - La clé secrète ne doit pas être:
    - transmise sur le réseau
    - Stockée sur l'équipement en clair
  - La taille de la clé est fixe
    - Généralement indépendante de la taille du message
    - Limitée en taille par la législation en cas de confidentialité
  - Une clé différente pour chaque couple
    - Absence de gestion de clés
    - N x (N – 1) / 2

Eléments de cryptographie

- Algorithmes de chiffrement Asymétrique
  - Deux clés (k_1 et k_2): si l'une chiffre l'autre déchiffre
  - Il existe une relation unique entre les deux clés (k_1 et k_2)
    - Une clé sera rendue publique et publiée dans un annuaire: **clé publique**
    - L'autre clé restera privé et connu et détenu par une et une seule entité: **clé privé**



Eléments de cryptographie

- Algorithmes de chiffrement Asymétrique
    - La clé privée est du domaine du privé

    - La clé privée ne doit jamais être:
        - divulguée à un tiers
        - stockée en clair sur un support quelconque
        - échangée au travers du réseau en chiffrée ou en clair

Eléments de cryptographie

- Algorithmes de chiffrement Asymétrique
  - Algorithme RSA (R.Rivest,A.Shamir,L. Adleman)
    - Défini en 1977 breveté par le MIT(expiration en 2000)
    - Basé sur la factorisation des nombres premiers
    - Commercialisé par la société RSA
    - Le plus déployé parmi les algorithmes asymétrique
      - Obligatoire dans plusieurs protocoles (SSL/TLS,PGP, SET,..)
      - Intégré à presque toutes les cartes de paiement
  - Algorithme El Gamal
    - Réponse à un appel du NIST
      - pour échapper à la patente de RSA
      - Basé sur les logarithmes discrets
        - Complexité comparable à RSA

Eléments de cryptographie

- Algorithmes de chiffrement Asymétrique
  - Clé Publique est :
    - le couple (e,n)
  - Clé Privée est :
    - le couple (d,n)
  - Soit M le message clair et C le message chiffré.
  - Pour chiffrer M, on calcule :
    - $C = M^e$ modulo n.
  - Pour déchiffrer on calcule :
    - $M = C^d$ modulo n.

Eléments de cryptographie

- Algorithmes de chiffrement Asymétrique

  - Pour former les couples *(e,n)* et *(d,n)*.
    - On choisit au hasard 2 gds. nombres premiers *p* et *q*.
    - On calcule $n = p.q$
    - On pose $j = (p-1).(q-1)$
    - $\varphi(n) = j$
    - On sélectionne e tel que :
      - *e* et *j* soient premiers entre eux avec $1 < e < j$
    - On calcule *d* tel que :
      - $e.d = 1 \bmod j$ (*e* et *d* sont inverses l'un de l'autre modulo *j*)

Eléments de cryptographie

- Algorithmes de chiffrement Asymétrique

  - Exemple:
    p = 3, q =11
    n = p.q = 3 x 11 = 33
    j = (p-1)(q-1) = 2 x 10 = 20
    Pour e = 3, d = 7
    Car (e.d = 1 mod j): 3 x 7 = 1 mod j

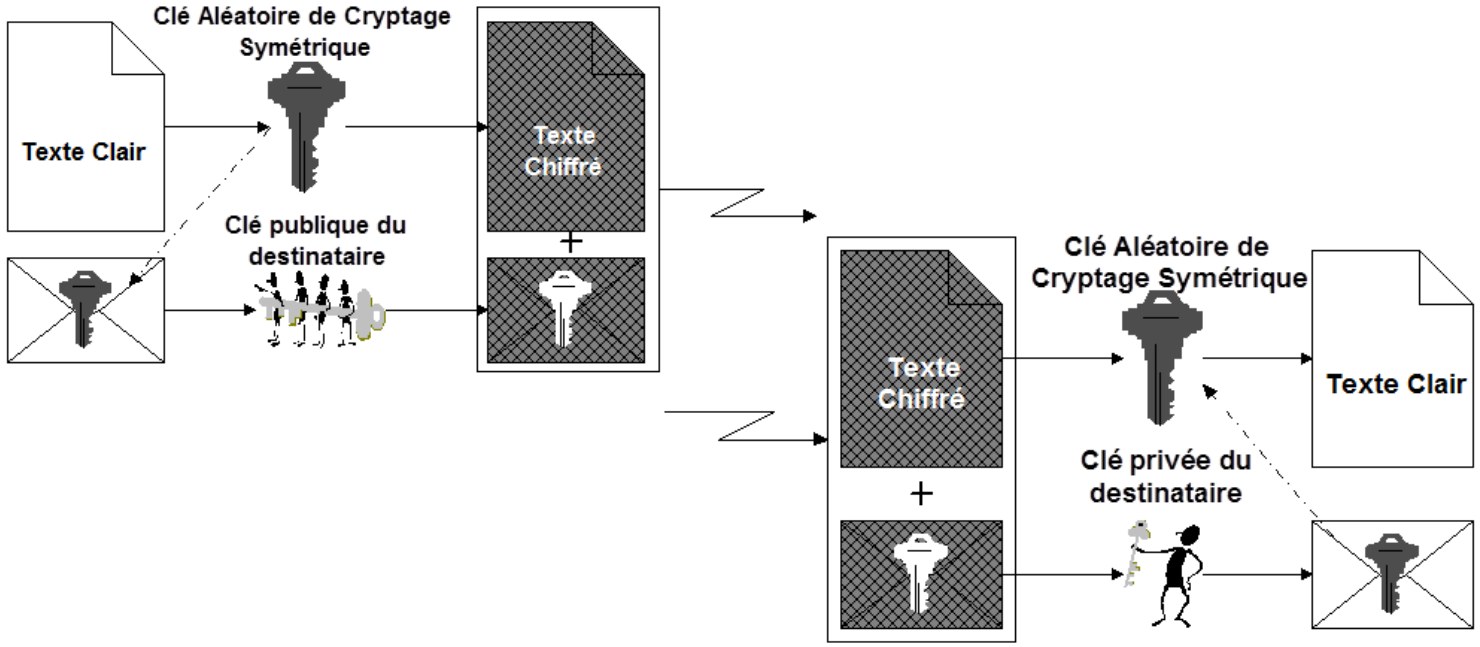
    Pour un message: M = 29

    Chiffrement:
    $Y = M^e \bmod n = 29^3 \bmod 33 = 2$

    Déchiffrement:
    $Y^d \bmod n = 2^7 \bmod 33 = 29$

Eléments de cryptographie

- Utilisation conjointe du chiffrement symétrique et Asymétrique



Eléments de cryptographie

- Fonction de hachage
  - H (M) = C
    - M est de taille quelconque
    - C est de taille fixe (16 ou 20 octets)
      - C est appelé condensât, ou empreinte, ou fingerprint, ou message digest
  - Fonction à sens unique
    - Si H (M_1) = C_1 ,
      - il est très difficile de trouver :
        M_2 différent de M_1 tel que H (M_2) = C_1
  - Usage : checksums, « intégrité »

Eléments de cryptographie

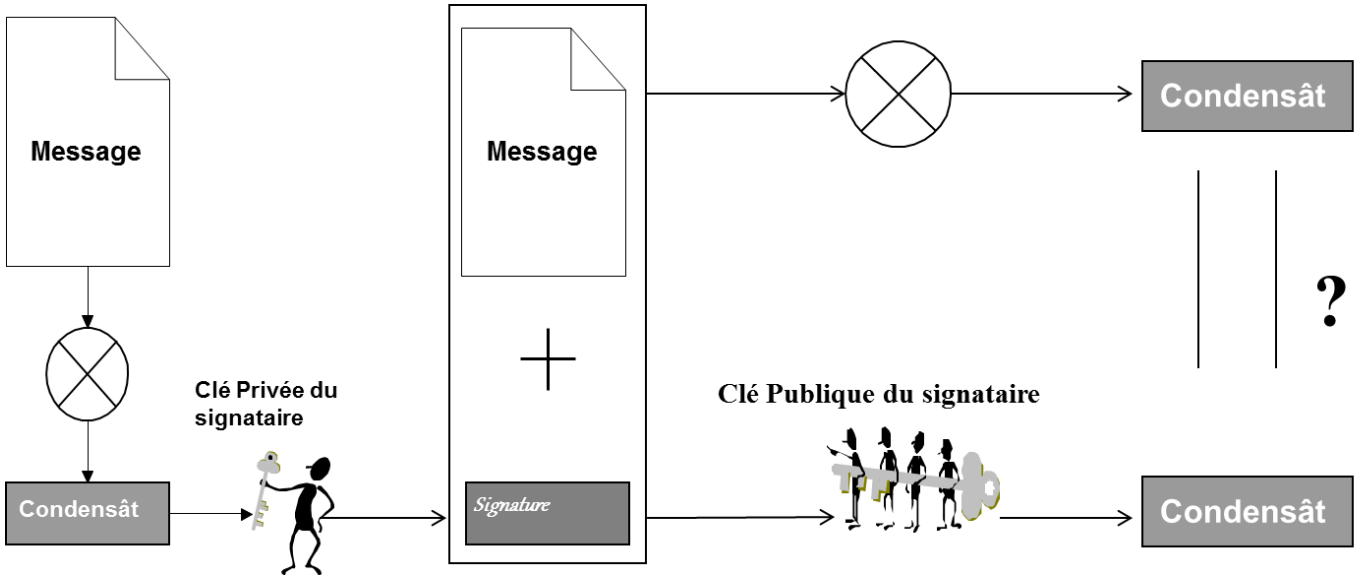
- Fonction de hachage avec clés
  - Hmac avec trois paramètres:
    - Fonction de hachage
    - Clé secrète
  - Message de taille quelconque
  - Fonction à sens unique
- Résultat :
  - mac (Massage Authentication code)
  - Usage : intégrité, authentification
  - Ce n’est pas une signature!

Eléments de cryptographie

- Fonction de Hachage sans clé
    - A la base des fonctions de hachage à clés
    - Signatures numériques (avec le chiffrement asymétrique)
    - mot de passe: stockage des hachés
  - Message Digest : MD2, MD4 et MD5.
    - Développé par Ron Rivest pour la RSA Security
    - **http://www.ietf.org/rfc/rfc1319.txt ;rfc1320.txt et rfc1321.txt.**
  - RACE Integrity Primitives Evaluation Messages Digest
    - **RIPEMD**-128 et RIPEMD-160.
    - Développé par H. Dobbertin, A. Bosselaers et B. Preneel
    - http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html
  - Secure Hash Algorithm SHA1 (standard SHS).
    - Développé par le NIST en 1995; ANSI X9.30.
    - http://www.itl.nist.gov/fipspubs/fips180-1.htm

Eléments de cryptographie

- La signature numérique



Eléments de cryptographie

- La signature numérique:
    - permet de mettre en œuvre les services:
        - Intégrité du message
        - Authentification
        - Identification
        - Non-répudiation

Eléments de cryptographie

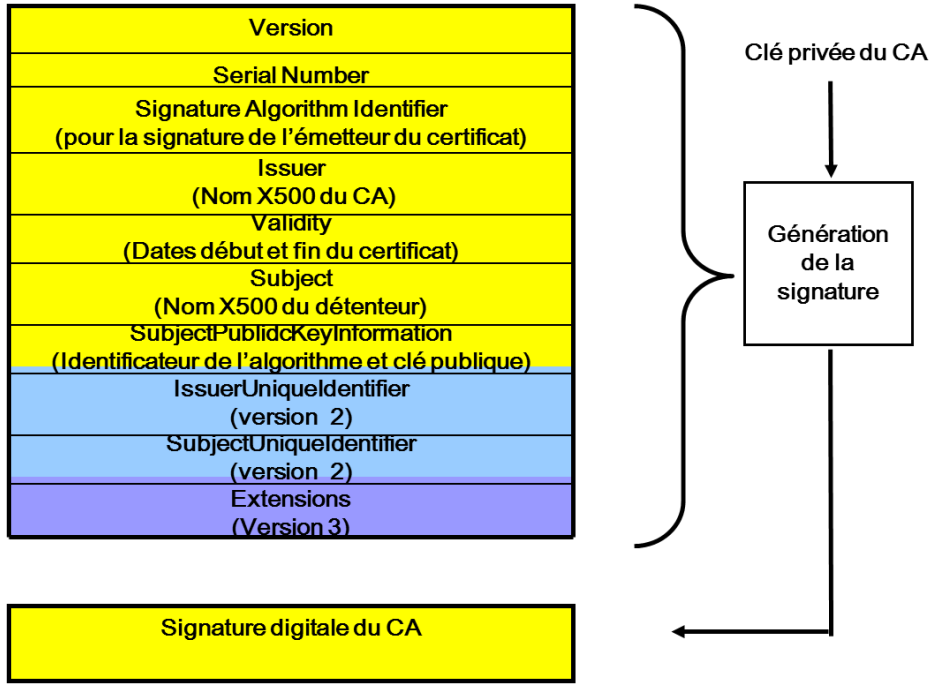
- Les Certificats X.509
  - « Certificat » = relève d'une autorité ou institution
  - Le contenu = information « authentique »
  - Mise en place d'un état de confiance en présence d'un certificat
  - Dico: « Acte écrit qui rend témoignage de la vérité d 'un fait, d'un droit »
  - Présence d'une autorité « reconnu » qui atteste de la véracité du contenu.
  - Certificat = document « signé »

Eléments de cryptographie

- Les Certificats X.509
  - Standard:
    - ITU-T X.509(03/2000), ou ISO/IEC 9594-8
      - Certificats de clé publique et d'attribut
    - RFC 3280: (définition de profil fonctionnel basé sur X509)
  - Versions successives:
    - 1988 : v1
    - 1993 : v2 = v1 + 2 nouveaux champs
    - 1996 : v3 = v2 + extensions

Eléments de cryptographie

- Les Certificats X.509
  - Structure de données permettant de lier différents éléments au moyen d'une signature
    - Le sujet ,la clef, l'émetteur du certificat, conditions de validité,...



Eléments de cryptographie

- Les Certificats X.509

Certificat ::= SEQUENCE {

version[0] Version DEFAULT v1,

serialNumber CertificateSerialNumber,

signature AlgorithmIdentifier,

issuer Name,

validity Validity,

subject Name,

subjectPublicKeyInfo SubjectPublicKeyInfo,

issuerUniqueIdentifier[1] IMPLICIT UniqueIdentifier OPTIONAL,

-- si ce composant est présent, la version doit être v2 ou v3

subjectUniqueIdentifier[2]IMPLICIT UniqueIdentifier OPTIONAL, *-- si ce composant est présent, la version doit être v2 ou v3*

extensions[3] Extensions OPTIONAL -- si ce composant est présent, la version doit être v3 --

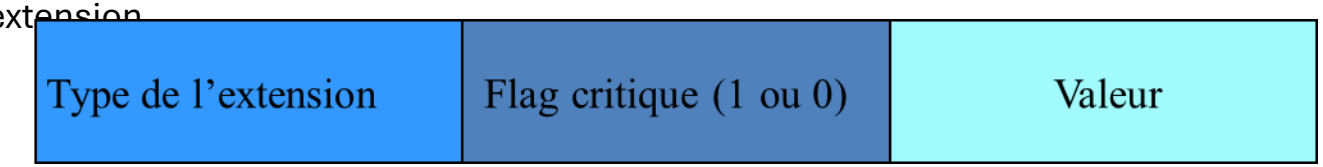
}

Eléments de cryptographie

- Les Certificats X.509
  - Le but initial des certificats est de lier:
    - identité et clé publique par la signature d'un tiers de confiance
  - Pour couvrir des services plus étendus
    - Nécessaire d'associer d'autres informations à la clé publique
  - L'extension consiste à ajouter de nouveaux champs aux certificats
  - L' extension est défini dans ITU-T Rec. X.660 et ISO/IEC 9834-1
  - Des extensions sont standardisées,
    - possibilité de définir des extensions spécifiques
  - Si l'application ne supporte pas une extension critique, elle abandonne le certificat.

Eléments de cryptographie

- Les Certificats X.509
  - Structure de l'extension



- Type est unique pour chaque extension: le type est un type de base ASN1 Object Identifier (OID)
- Avec un flag critique
  - Si l'application:
    - ne supporte pas cette extension, elle refuse le certificat
    - supporte cette extension et la valeur de l'extension est conforme à l'application elle l'accepte sinon elle le rejette
- Avec un flag non critique
  - Si l'application
    - ne supporte pas cette extension, l'extension est a supporte pas cette extension abandonnée mais le certificat accepté
- La valeur est conforme au type de l'extension

Eléments de cryptographie

- Les Certificats X.509
  - Extensions sur:
    - le nomage de l’objet et du signataire
    - les clés publiques/privées
    - la révocation
    - la politique de certification
    - le rôle
    - Autres … logo (RFC 3709: Internet X.509 Public Key Infrastructure:Logotypes in X.509 Certificates)

Eléments de cryptographie

- Les Certificats X.509

- KeyUsage : usage de la clé publique certifiée

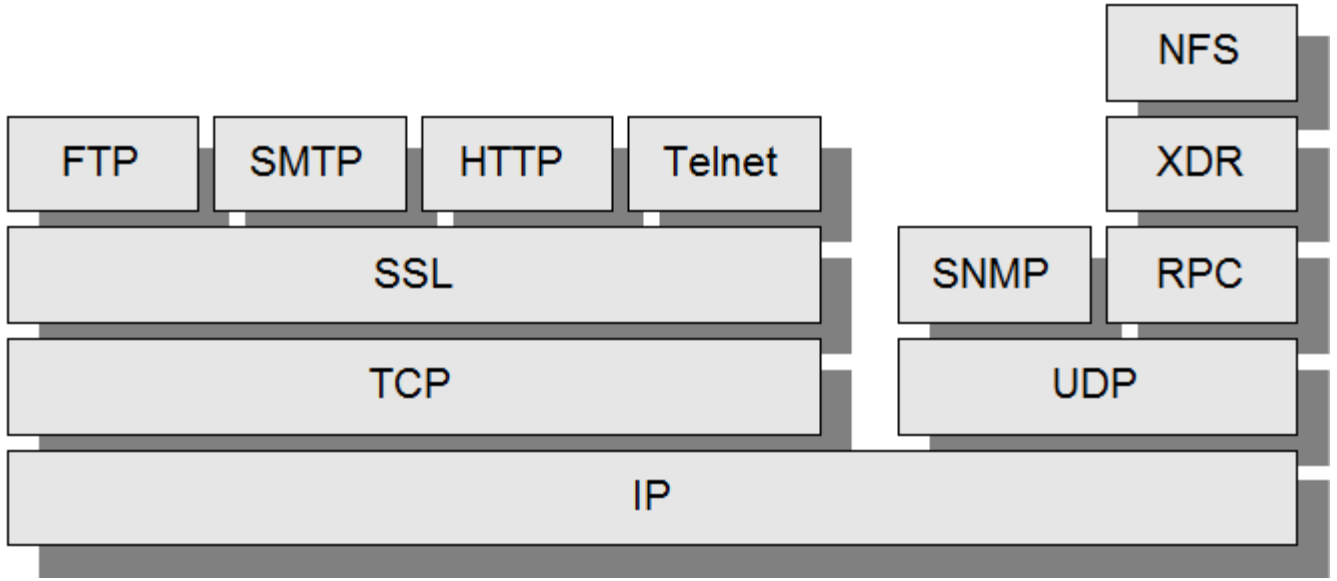
  DigitalSignature
  NonRepudiation
  KeyEncipherment
  DataEncipherment
  keyAgreement
  keyCertSign
  CRLSign
  encipherOnly
  decipherOnly

Le protocole SSL/TLS

- SSL / TLS
  - SSL défini par *netsacpe* et intégré au browser
  - Première version de SSL testée en interne Première version de SSL diffusée : V2 (1994)
  - Version actuelle V3
  - Standard à l ’IETF au sein du groupe Transport Layer Security (TLS)
  - Standard au sein du WAP Forum Wireless Transport Layer Security (WTLS)

Le protocole SSL/TLS

- SSL / TLS



Le protocole SSL/TLS

- SSL / TLS

Protocole sécurisé	Port	Protocole non sécurisé	Application
HTTPS	443	HTTP	Transactions requête-réponse sécurisées
SSMTP	465	SMTP	Messagerie électronique
SNNTP	563	NNTP	News sur le réseau Internet
SSL-LDAP	636	LDAP	Annuaire X.500 allégé
SPOP3	995	POP3	Accès distant à la boîte aux lettres avec rapatriement des messages

Le protocole SSL/TLS

- SSL / TLS

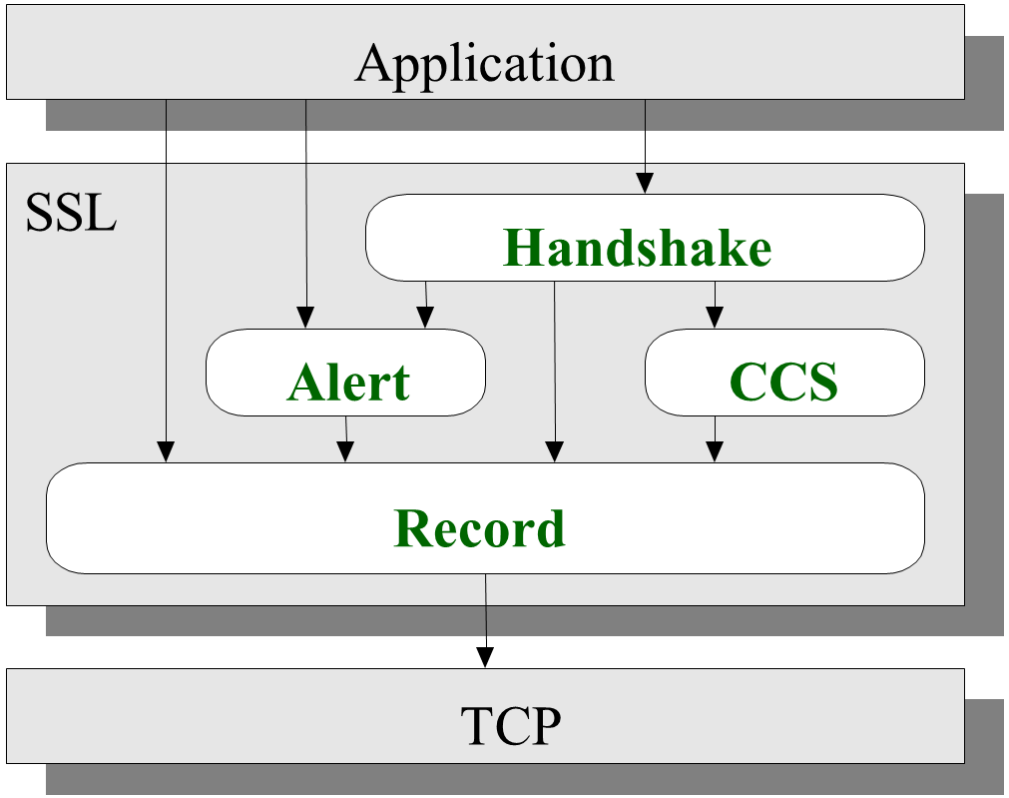
Protocole sécurisé	Port	Protocole non sécurisé	Application
FTP-DATA	889	FTP	Transfert de fichiers
FTPS	990	FTP	Contrôle du transfert de fichiers
IMAPS	991	IMAP4	Accès distant à la boîte aux lettres avec ou sans rapatriement des messages
TELNETS	992	Telnet	Protocole d'accès distant à un système informatique
IRCS	993	IRC	Protocole de conférence par l'écrit

Le protocole SSL/TLS

- SSL / TLS
  - Authentification
    - Serveur (obligatoire), client (optionnel)
    - Utilisation de certificat X509 V3
    - A l'établissement de la session.
  - Confidentialité
    - Algorithme de chiffrement symétrique négocié, clé générée à l'établissement de la session.
  - Intégrité
    - Fonction de hachage avec clé secrète : hmac(clé secrète, h, Message)
  - Non Rejeu
    - Numéro de séquence

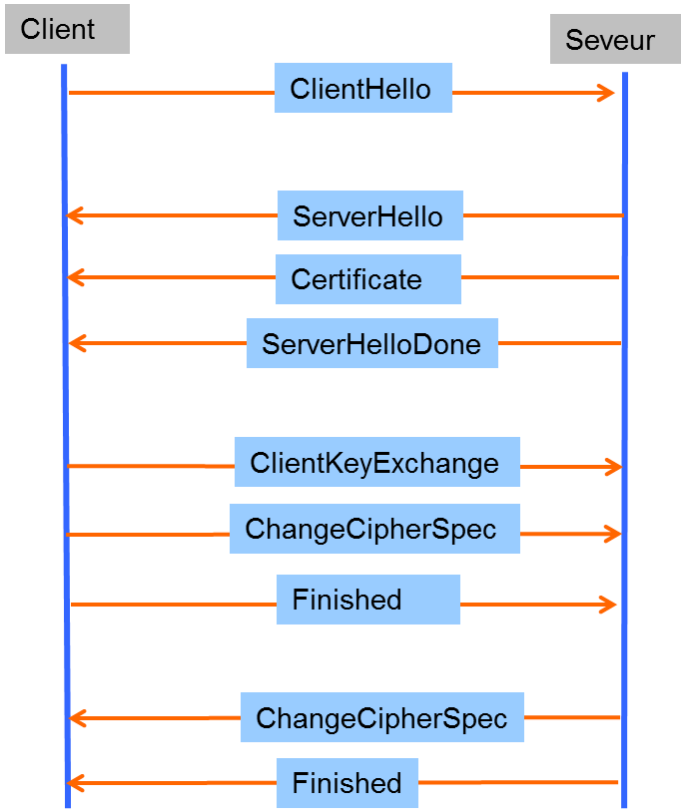
Le protocole SSL/TLS

- SSL / TLS



Le protocole SSL/TLS

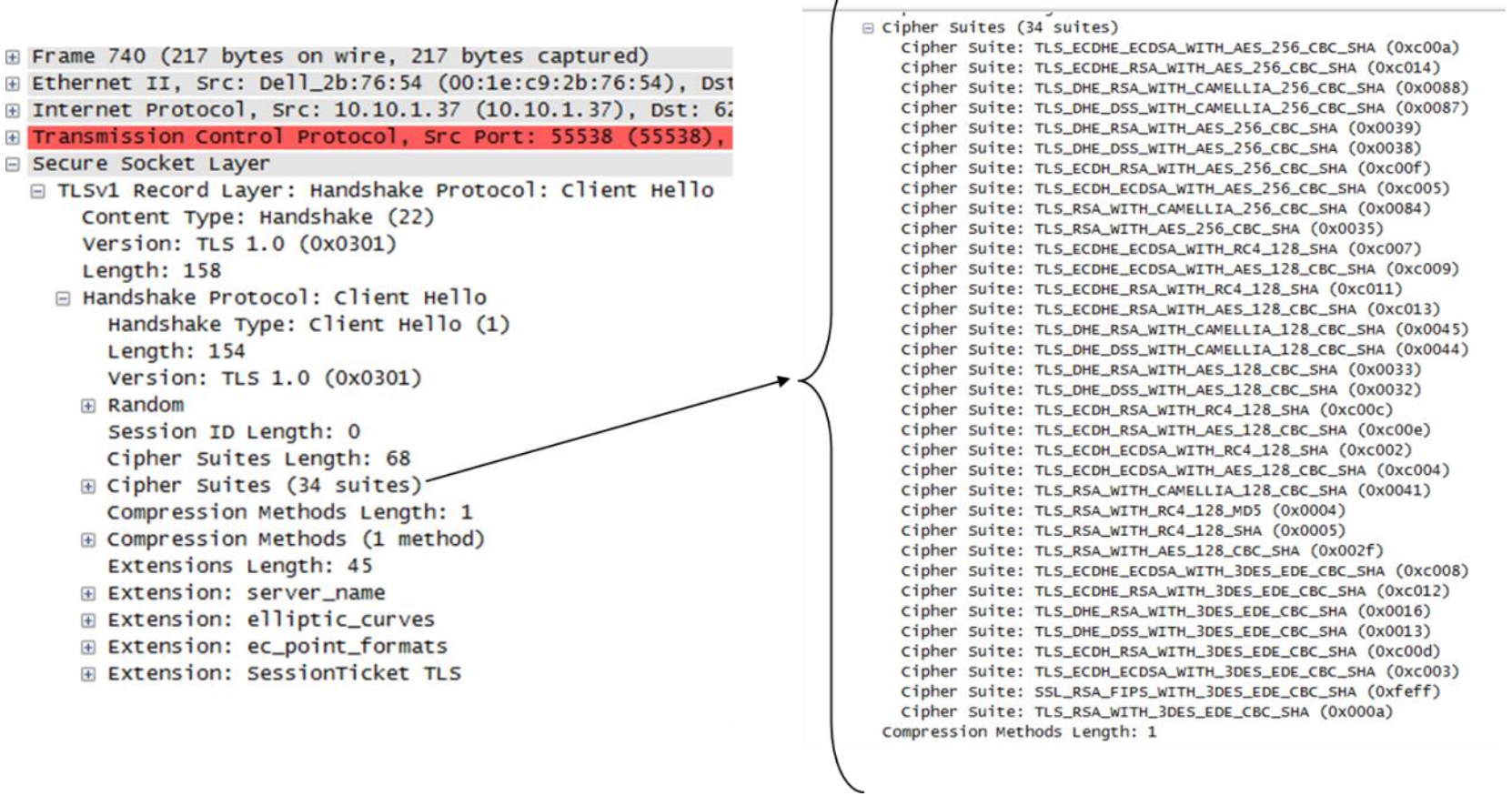
- SSL / TLS Handshake Protocole



- *ClientHello* : ce message contient la version de SSL, un nombre aléatoire permettant de générer les clés secrètes, et l'ensemble d'algorithmes proposés : DH, RSA, 3DES
- *Server Hello* : dans ce message le serveur choisit les algorithmes proposés par le client et choisit la longueur de clés. Ainsi, il génère un nombre aléatoire permettant de générer les clés secrètes
- *Certificate* : le navigateur du client vérifie immédiatement la validité du certificat
- *ServerHelloDone : Ok => à ce moment le client peut vérifier le certificat du serveur et échanger les clés*
- *ClientKeyExchange :* le client génère la clé pre-master et le chiffre par la clé publique du serveur. Enfin, il envoie le message chiffré au serveur
- *ChangeCipherSpec* : le client informe le serveur que tous les messages suivants vont être chiffrés par la clé symétrique échangée dans le message d'avant
- *Finished* : le client envoie un message chiffré par la nouvelle clé pour vérification
- Le serveur fait la même procédure pour vérifier la bonne utilisation de la clé

Le protocole SSL/TLS

- SSL / TLS: Handshake (Client Hello)



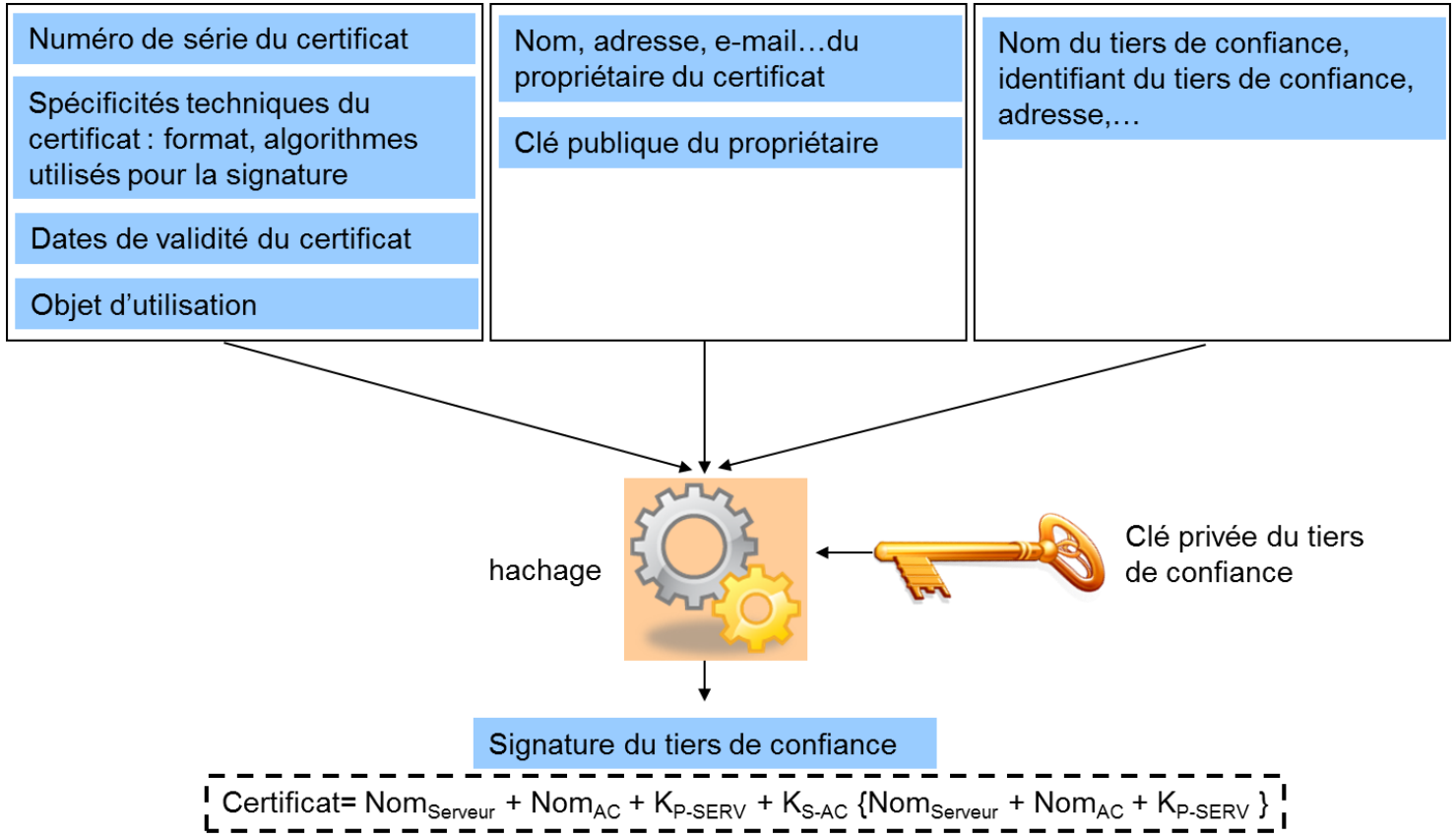
Le protocole SSL/TLS

- SSL / TLS: handshake (Server Hello)

```
741 17.003003   62.161.94.179        10.10.1.37          TLSv1    Server Hello, Certificate, Server Hello Done
⊞ Frame 741 (1058 bytes on wire, 1058 bytes captured)
⊞ Ethernet II, Src: Cisco_d2:49:3f (00:1f:6c:d2:49:3f), Dst: Dell_2b:76:54 (00:1e:c9:2b:76:54)
⊞ Internet Protocol, Src: 62.161.94.179 (62.161.94.179), Dst: 10.10.1.37 (10.10.1.37)
⊞ Transmission Control Protocol, Src Port: https (443), Dst Port: 55538 (55538), Seq: 1, Ack: 164, Len: 1004
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 999
    ⊟ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 70
        Version: TLS 1.0 (0x0301)
      ⊞ Random
        Session ID Length: 32
        Session ID: 08010000EF91A59A714BD60A7F42FCA1FFE867C1207CCFE9...
        Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
        Compression Method: null (0)
    ⊟ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 917
        Certificates Length: 914
      ⊞ Certificates (914 bytes)
    ⊟ Handshake Protocol: Server Hello Done
        Handshake Type: Server Hello Done (14)
        Length: 0
```

Le protocole SSL/TLS

- SSL / TLS: handshake (certificate)



Authentification des utilisateurs

- Principes:
  - Authentification consiste à:
    - Confirmer de la véracité de l'identité ou d'un élément spécifique à une entité déclarée
  - Authentification
    - Simple: une seule entité est authentifiée
    - Mutuelle: l'ensembles des entités sont authentifiées
  - Authentification
    - Souple: basée sur une secret partagé
    - Forte: basée sur un élément privé
- Peut être utilisée comme élément de preuve

Authentification des utilisateurs

- Principes:
  - Authentification peut être basée:
    - Sur des éléments logiques (mot de passe, certificat, …)
    - Ou sur des éléments physiques associés à l'entité à authentifier (biométrie)
  - Les mots de passe sont les plus usuels actuellement:
    - Gestion relativement simple
    - Stocké en clair ou « chiffré » du côté serveur
    - Peuvent être également stockées du côté client en clair
      - Stockées de manière opaque et non gérable
  - Authentification par certificats:
    - Nécessitent une infrastructure

Authentification des utilisateurs

- Principes:
  - Authentification par mot de passe en clair
    - Échange en clair sur le réseau
    - Peut être stocké en clair du côté serveur selon les applications
      - Telnet, Ftp, POP, HTTP basic, bases de données (SQL), …
    - Parfois commun à plusieurs services
    - Exemple: Protocole PAP (Password Authentication Protocol)
  - Authentification par mot de passe avec challenge
    - Le mot de passe ne transite pas durant l'échange
    - Est toujours stocké en clair du côté serveur
    - Rarement commun à plusieurs services
    - HTTP digest
    - Exemple: protocole CHAP (Challenge Authentication Protocol)

Authentification des utilisateurs

- Principes:
  - Authentification
    - Au niveau SSL/TLS
      - Par certificat
    - Au niveau HTTP
      - http basic et http digest
    - Au niveau applicatif
      - Par mot de passe en clair
    - Au niveau réseau (IPsec)
      - PSK, Kerberos, Certificat
    - Authentification la plus usuelle
      - Du serveur au niveau SSL/TLS avec un certificat
      - Du client au niveau applicatif avec mot de passe en clair

Authentification des utilisateurs

- HTTP BASIC et DIGEST

  - Défini dans le RFC2617
  - Usuel dans SIP pour la VoIP
  - **HTTP Basic**
    - Le client fait une requête http vers le serveur
    - Le serveur lui retourne un message d'erreur (statuts 401) et lui demande de s'authentifier avec le mode basic
    - Le client refait sa première requête avec des entêtes en plus incluant son identifiant (login) et son mot de passe, le tout en clair
    - Le serveur lui retourne la ressource si les paramètres fournis ont été validés

Authentification des utilisateurs

- HTTP BASIC et DIGEST
  - **HTTP Digest**
    - Le client fait une requête http vers le serveur
    - Le serveur lui retourne un message d'erreur (statuts 401) et lui demande de s'authentifier avec le mode digest, le serveur lui renvoie un nonce (nombre aléatoire)
    - Le client refait sa première requête avec des entêtes en plus incluant son identifiant (login) et le résultat du haché de son mot de passe combiné avec le nonce et éventuellement d'autres paramètres (le mot de passe ne transite pas en clair)
    - Le serveur lui retourne la ressource si les paramètres fournis ont été validés

Authentification des utilisateurs

- HTTP BASIC et DIGEST

Réponse du serveur à une requête du client nécessitant l'authentification du client:

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

Réponse du client au message précédent:
Authorization:

Digest username="Mufasa",
realm="testrealm@host.com",
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
uri="/dir/index.html",
qop=auth,
nc=00000001,
cnonce="0a4f113b",
response="6629fae49393a05397450978507c4ef1",
opaque="5ccc069c403ebaf9f0171e9517f40e41"

Authentification des utilisateurs

- Simple Authentication and Security Layer SASL

  - Défini dans le RFC4422
  - Pour découpler l'authentification des applications
  - Permet la mise en œuvre d'architecture flexible
  - De type req/rep, SASL supporte une API pour le support de plusieurs méthodes d'authentification
    - External, Anonymous, Plain, Digest, Skey, OTP, NTLM, GSSAPI-Kerberos

Authentification des utilisateurs

- S/KEY un OTP de type logiciel
  - RFC1760
  - Initialisation par une pass phrase du client/serveur
  - Le serveur transmet un « un nombre »
  - Le client génère un mot de passe en utilisant MD5, la passe phrase du client, le nombre
  - Le client transmet le résultat en clair
  - Ce résultat est utilisé une et une seule fois pour l'authentification du client
- OTP défini dans le RFC1938 est une évolution de S/KEY qui est une marque déposée de la compagnie américaine Bellecore

- SECURID, ActivCard, CryptoCard solutions de type hard

Configuration des systèmes et des logiciels

- Sécurisation des Réseaux
- Sécurisation des systèmes
- Sécurisation de Windows
- Sécurisation de Linux
- Sécurisation des Services
- Sécurisation des Serveurs Web
- Sécurisation des Applications Web

Sécurisation des Réseaux

- Contrôle d’accès entre applications Web et les réseaux Internet
  - Firewall, routeur, etc.
- Firewall ou routeur configuré pour n'autoriser que le trafic destiné à l’Application Web:
  - HTTP et SSL/TLS
  - Limité le nombre de TCP Syn en entrée
- Firewall ou routeur configuré pour n'autoriser que le trafic sortant issu de l'application Web:
  - Pas de TCP Syn en sortie
- Équilibreur de charge configuré pour ne pas divulguer des informations internes sur les réseaux
- Déployer un IDS pour renforcer la détection d’intrusion
- Journalisation des logs
- Planifier une politique de scans pour la détection d’intrusion

Sécurisation des Systèmes

- Après son installation, toute application est globalement « unsecure »
- Les règles à suivre:
  - Installer les éléments nécessaires
  - Pas ceux qui peuvent l'être !!
  - Ne pas installer des exemples ou de la documentation
  - Appliquer les derniers patches (Service Pack, Hot-Fix, Upgrades…)
  - Utiliser des mots de passe robustes
  - Mettre en place une gestion des mots de passe (durée de vie, enregistrement, renouvellement, logs, …)
- Analyser les fichiers de configuration par défaut :
  - Compte root/administrateur
  - Accès et partages administratifs
  - Appliquer des permissions d'accès sur les fichiers au niveau du serveur
  - Accès linux single…

Sécurisation de windows

- Appliquer le dernier Service Pack et les Hot-fixes post Service Pack
  - http://www.microsoft.com/windows/security/
- Compte Administrateur par défaut (local)
  - Le renommer
  - Interdire son utilisation par le réseau
- Compte d'administration courant
  - Créer un compte « superuser » par exemple
  - Le placer dans le groupe « administrateurs »
  - Activer la sécurité des comptes
- Gestion des mots de passe
  - Lockout
  - Audit succès / échecs de connexion
- Arrêter tous les services inutiles
  - Méthode itérative
    - Mode manuel des services potentiellement inutiles
    - Reboot
    - Mode désactivé pour les services effectivement inutiles
- Permissions sur le système de fichiers et la base de registre
- Désactivation des « null sessions »

Sécurisation de Linux

- Partitionnement, au minimum :
  - / - 1 Go
  - /var - le reste
  - partition swap - 2 x RAM
- N ’installer que les packages nécessaires
  - rpm -e ou rpm –i
- Veille sécurité sur bulletins officiels RedHat
  - http://www.redhat.com/support/errata
  - Packages (rpm -Uvh)
  - Ne pas utiliser les autorpm
- Patcher le noyau
  - http://www.redhat.com/support/docs/howto/kernelupgrade/
  - kernel-upgrade.html

Sécurisation des Services

- Super-serveur : Internet Daemon
  - Invalider les services en commentant les lignes dans /etc/inetd.conf
  - Changer les droits: chmod 600 /etc/inetd.conf
- Services applicatifs
  - dans /etc/rc.d
  - Démarrage à contrôler avec ntsysv ou chkconfig
  - Surveiller les services RPC, DNS, NFS, Serveur X
- Contrôler et éliminer les services inutiles.
  - Limiter les ports d'écoute :
    - netstat –lnptu
- Utiliser au maximum des canaux chiffrés (redirection de ports avec SSH)
  - Remplacer telnet par SSH.