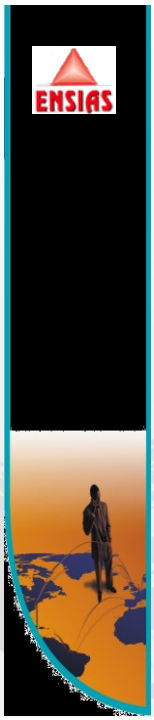


Sécurité Applicative

3A - SSI

Pr. El Bakkali Hanane

Sommaire du cours

I. Introduction

1. *Pourquoi sécuriser?*
2. *Où et quoi sécuriser?*
3. *Comment sécuriser?*
4. *Politique de sécurité*

I.1. Pourquoi sécuriser ?

- Les SI sont de plus en plus considérés comme les centres névralgiques des organismes;
- Parfois, le SI est au cœur de l'activité de l'organisme;
- Les SI sont de plus en plus connectés à Internet;
- Il existe différentes motivations pour attaquer un SI, dont :
  - Bénéfices financiers (vol de n° de carte de crédit, espionnage industriel, nuisance à l'image de marque profitant aux concurrents, ..);
  - Satisfaction personnelle (plaisir/jeu, fierté maladive, concurrence entre Hackers, etc);
  - Vengeance (salarié licencié et/ou sous estimé, ...)
  - Convictions politiques et/ou idéologiques (partisans d'un parti, terroristes, ...);
  - Espionnage d'état.

I.1. Pourquoi sécuriser ?

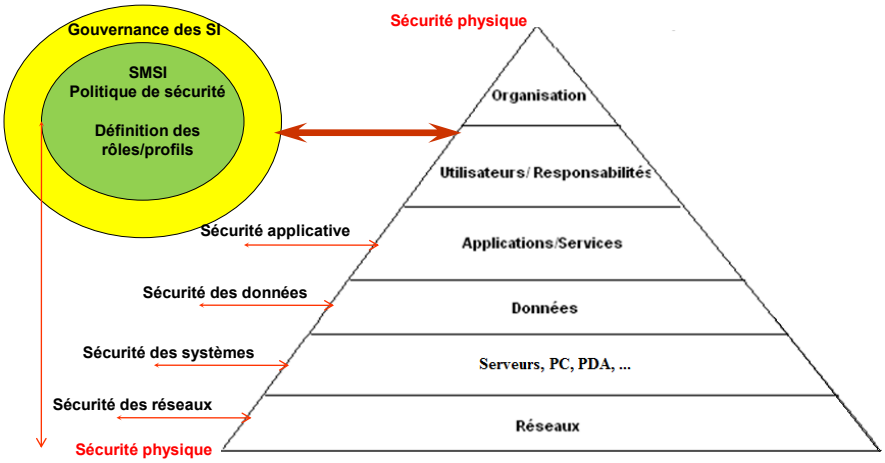
- Suivre les tendances: la sécurité est à la mode !
  - Le marché de la sécurité est en pleine croissance;
  - **Les budgets 'Sécurité IT':** une augmentation de 24% en 2015 **(**The Global State of Information Security Survey 2016)
- Les statistiques montrent que les attaques (déclarées ou découvertes) ont tendance à la fois à croître et à devenir plus efficaces et qu'elles touchent même les grandes sociétés des TIC. Par exemple, en 2015, 38% plus d'incidents détectés qu'en 2014 (même source)
- Toutes les attaques exploitent des faiblesses de sécurité au niveau du SI avec une 'migration' vers le niveau **application** ;
- Les pertes dues à une attaque peuvent être très graves (rien qu'en termes de temps perdu, elles peuvent être considérables)
- Enfin, la cybercriminalité n'est pas encore bien cernée par les instances judiciaires, ce qui rend le recours à la justice peu fructueux.

I.2. Où et quoi sécuriser ?

- Les Besoins en termes de sécurité se situent à plusieurs niveaux (cela dit, les frontières entres ces différents niveaux sont loin d'être bien nettes) :
  - Sécurité des locaux, des infrastructures et des équipements (physique);
  - Sécurité des réseaux;
  - Sécurité des **applications** et des systèmes d'exploitation;
  - Sécurité des données (exp. BD, Datawarehouses, BigData …).

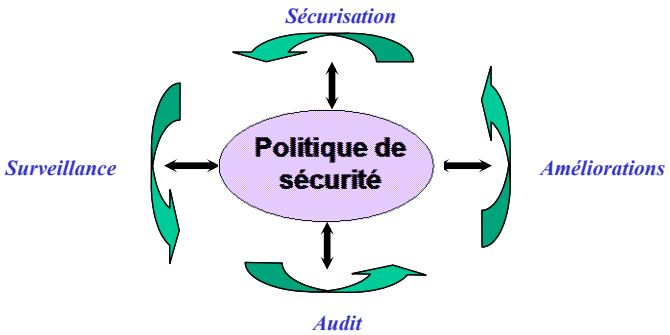
Il faut donc utiliser des mesures et des outils de sécurité distincts pour assurer la sécurité à chaque niveau, en vue de sécuriser globalement tout le SI.

I.2. Où et quoi sécuriser ?



I.3. Comment sécuriser ?

➔ *Cycle de la sécurité (Approche : PDCA):*



➔ *≈Approche : PDCA*

I.4. Politique de sécurité

☞ Elle doit être la base, le point de départ de tout projet de sécurité;

☞ L'implication des hauts responsables dans l'élaboration d'une PS est plus que souhaitable; elle est indispensable pour la réussite du projet;

☞ Une PS est une structure autour de laquelle un organisme construit tous les aspects de sécurisation de son SI;

☞ Elle doit définir les règles représentant les accès acceptables aux ressources du SI : politique de contrôle d'accès;

☞ Il existe diverses normes (famille **ISO 27000 et notamment l'**ISO/IEC 27034-1:2011 qui concerne la sécurité applicative) et méthodes (bonnes pratiques) qui peuvent servir de guide à l'élaboration d'une politique de sécurité (exp: Marion, Méhari pour l'analyse des risques).

☞Différents modèles formels de politique de sécurité (surtout de contrôle d'accès) existent dans la littérature (exp. Bell-Lapadula, Clark et wilson, RBAC, …)

I.4. Politique de sécurité :

☞Il est recommandé qu'elle traite d'abord les points suivants:

- **Diagnostic de l'existant**:
  - Recensement et classification des ressources à protéger;
  - Identification de l'infrastructure Réseau;
  - Inventaire des outils de sécurité existants;
  - Aspects organisationnels.
- **Analyse des risques** :
  - Estimation des vulnérabilités;
  - Identification des menaces éventuelles;
  - Estimation des pertes directes et indirectes pouvant être causées par chaque menace;
  - Définition des priorités concernant les ressources à protéger.

I.4. Politique de sécurité :

➢ **Définition des rôles (qui est responsable de quoi):**

- Existence ou non d'un RSSI;
- Externalisation ou non de la sécurité (si oui: totale ou partielle);
- Hiérarchie de la sécurité:

  Par exemple: - RSSI à DSI à DG
  - RSSI à DG
  - RSSI à Direction métier
- Chaînage des responsabilités au sein de l'équipe Sécurité;
- Rôle (en termes de sécurité) de l'utilisateur final (salarié, client ou fournisseur).

Importance de la sensibilisation à la 'culture' de la sécurité !

I.4. Politique de sécurité :

➢ **Estimation du coût/ budget de la sécurité**:

- Le coût de démarrage doit être raisonnable par rapport aux pertes possibles à moyen terme;
- Le budget annuel de la sécurité (parfois difficile à séparer du budget informatique) doit être inférieur aux pertes probables pendant un an;
- Choix du niveau de sécurité à atteindre(en termes OSI: classe de sécurité) dépend du budget alloué;
- Choix fonctionnels et techniques, après:
  - ✓ Comparaison (technique et financière) entre différents outils et technologies disponibles sur le marché;
  - ✓ Formation éventuelle du personnel (du DG à l'utilisateur final);
  - ✓ Sollicitation éventuelle de prestataires de service externes.

I.4. Politique de sécurité :

☞ Elle doit offrir -en réponse aux points précités- une sorte de plan directeur de sécurité qui contient le détail des procédures suivantes:

- **Procédures de protection**, dont:
  - Politique de contrôle d'accès qui comprend les procédures d'accès aux différentes ressources (système, réseau, données, etc);
  - Mécanismes de contrôle et de neutralisation des virus;
  - Mécanismes de protection des informations confidentielles;
  - Procédures de sauvegarde;
  - Procédures de configuration et de mise à jour des outils installés et des correctifs correspondants.

I.4. Politique de sécurité :

- **Procédures de détection**, dont:
  - Mécanismes de surveillance des serveurs (données, messagerie, Web, etc.) ;
  - Mécanismes de surveillance des activités réseaux;
  - Procédures de 'lecture' et d'analyse des activités journalières (Exps: rapports de certains outils, messages d'alertes, tableaux de bord, etc).
- **Procédures de réaction** (juste après la détection de l'incident), dont:
  - Procédures de réaction à une attaque par pirate (différents scénarios);
  - Procédures de réaction à une attaque virale;
  - Procédures de réaction à des incidents de force majeure.

I.4. Politique de sécurité :

- **Procédures de redressement** (réaction à court et moyen terme), dont:
  - Plan de gestion de crise ;
  - Plan de continuité des activités.
  - Contrats d'assurances;
  - Contrats de maintenance;
  - Procédures judiciaires.

Conclusion : Importance d'une bonne politique de sécurité !

Sommaire du cours

II. Enjeux de la sécurité applicative

Sommaire

II.1. Risques au niveau applicatif

- **Applications à risque :**
  - Applications incontournables :
    - ✓ Messagerie électronique, DNS, etc.
    - ✓ Applications Web 'locales' et 'Publiques'
    - ✓ Progiciels,…
  - Applications en plein essor :
    - ✓ E-commerce, E-gouvernement, E-health, …
    - ✓ Applications mobiles, Internet des objets (IoT)
- **Menaces contre la sécurité des SI, ont tendance à migrer du niveau réseau au niveau applicatif (surtout les applications Web, les Web services et les applications mobiles ou sur Cloud) et plus récemment vers le niveau Transport (Vulnérabilité SSL : Heartbleed),**
- D'après 'Website Security Statistics Report 2015', En 2014, les applications sont –très probablement- moins protégées au niveau Transport (70% de risque) , fuite d'information (56%) et XSS (47%).

II.1. Risques au niveau applicatif

➢ **Principales attaques applicatives:**

- Attaques contre les applications Web
- Attaques contre les applications mobiles
- *Ingénierie sociale (pas vraiment une attaque applicative mais ..) et risques liés aux réseaux sociaux (Facebook, Twitter, …)*
- Spam, Spam et encore du Spam
- Attaques contre les serveurs DNS
- Attaques au niveau des sites Web
  - ✓ Le nombre de sites légitimes compromis dépasserait le nombre de sites créés par des cybercriminels.
  - ✓ Les sites de réseaux sociaux sont un trésor d'information pour les attaquants et un moyen d'acheminer les attaques (vol d'identité, phishing, distribution de vers, …).

II.2. Besoins de sécurité applicative

➢ **Besoins/Services de sécurité:**

- **Contrôle d'accès** aux applications et gestion des **autorisations**;
- **Authentification** des utilisateurs & serveurs (exp: sites Web) ;
- **Non-répudiation** des transactions par les utilisateurs;
- **Traçabilité** des actions des utilisateurs et des processus;
- **Intégrité** des données et **confidentialité** des données 'critiques' ;
- **Disponibilité** des applications/services/données;
- Résistance aux attaques applicatives (réduction des vulnérabilités).

➢ **Besoins annexes:**

- Convivialité (exp: nombre de clics ↘, nombre de mots de passe ↘) ;
- Performance (exp: temps de réponse ↘).

II.3. Moyens de sécurité actuels

- **Cryptographie à clé publiques/ à clés secrètes (quantique ?)**
- **Signature électronique**
- **Certificats de clés publiques**
- **Protocoles sécuritaires : SSL/TLS, IP-Sec, 3D-Secure, …**
- **Réseaux privés virtuels : VPN**

Infrastructures de clés publiques ou PKI

plus qu'une solution technique

II.3. Moyens de sécurité actuels

Signature électronique:

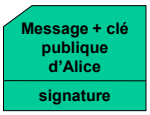
- **Fonctions d'une signature :**
  - Authentification du signataire
  - Acceptation présumée des termes du 'document' signé
  - Preuve en cas de litiges
- **Définitions :**
  - Électronique : Une donnée sous forme électronique qui est jointe ou liée logiquement à une unité de données et qui sert de méthode **d'authentification**. Elle peut être utilisée pour identifier le(s) signataire(s) d'un acte **juridique** accompli par voie électronique.
  - Numérique : données ajoutées à l'unité de données, ou une transformation cryptographique de cette unité permettant à un destinataire de prouver la source et l'intégrité de l'unité.

Signature électronique:

➢ **Définitions (suite):**

- Sécurisée (adoptée par le conseil des ministres du 13/4/06): une signature électronique qui doit :
  - ✓ être liée uniquement au signataire;
  - ✓ permettre d'identifier le signataire;
  - ✓ être créée par des moyens que le signataire puisse garder sous son contrôle exclusif;
  - ✓ être liée aux données auxquelles elle se rapporte, de telle sorte que toute modification ultérieure de ces données soit détectable.
- Sécurisée présumée fiable : Une signature sécurisée utilisant des moyens certifiés et des **certificats** qualifiés (Décret français du 30 mars 2001).

Signature électronique:



➢ **Problématique** :

- Une signature valide n'implique pas forcément que le message est bien signé par 'Alice', pour la simple raison qu'il se peut que **la clé publique** utilisée pour la vérification de la signature ne soit pas réellement la clé publique d'Alice mais celle d'une autre entité qui veut être prise pour Alice.
- Le vérificateur de la signature doit avoir la ***certitude*** que la clé publique est bien celle de Alice. Il lui faut donc un mécanisme permettant d'affirmer qu'une clé publique correspond ou non à une entité.

  → Si la *confidentialité* d'une clé publique n'est pas requise (au contraire, il faut la publier), son ***intégrité et son authenticité*** sont à préserver.

Certificat numérique :

- **Rôle/ utilisation :**
  - Un certificat de clé publique est un ensemble de données numériques dont le rôle est de lier une clé publique à son propriétaire pour qu'un utilisateur du certificat puisse croire à l'authenticité de la clé publique certifiée et donc à celle d'un document signé moyennant la clé privée associée.
  - Ce rôle ne peut être rempli que si :
    - ✓ le certificat est émis et signé par une tierce partie considérée 'de confiance' par le vérificateur du certificat et qui -par le biais de cette signature- affirme qu'elle est convaincue de l'authenticité de la clé publique qu'elle a certifiée.
    - ✓ Le vérificateur connaît de *manière sûre* la clé publique de la tierce partie afin de pouvoir vérifier sa signature apposée au certificat.

ENSIAS
II.3. Moyens de sécurité actuels

Certificat numérique :

- **Problématique :**
  - Si le vérificateur n'a pas pu obtenir 'directement' la clé publique de la tierce partie, il doit vérifier son authenticité, elle aussi, grâce à un autre certificat (de cette clé) signé par une autre tierce partie de confiance, et ainsi de suite.
  - On dit, dans ce cas, qu'il procède à la validation d'un chemin de certification.
    - ➡ Pour que ce chemin *aboutit*, il faut au moins que parmi ces tierces parties, il y en a une dont le vérificateur connaît, de manière sûre, la clé publique et ce, sans biais de certificat (on parle de 'Trust Anchor').

Certificat numérique :

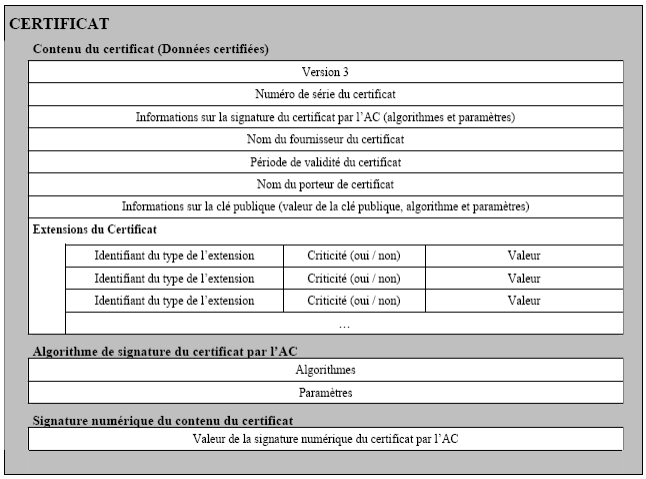
- **Description :**
  - En général, un certificat contient les informations suivantes :
    - ✓ Identifiant de l'émetteur du certificat (la tierce partie de confiance qui a signé le certificat) ;
    - ✓ Identifiant et/ou autre information de l'entité qui possède la clé publique à certifier (sujet du certificat) ;
    - ✓ La clé publique objet de certification ;
    - ✓ Autres informations (exp : date d'expiration).
    - ✓ **Signature** de l'émetteur du certificat
- **Formats les plus utilisés : X.509 V3, PGP**

Certificat numérique :

- **Certificat X.509 V3 (1/3):**
  - Il est le format le plus utilisé; il est défini par l'ISO/IEC JTC1 SC21 comme faisant partie des spécifications X.509 V3. :
    - ✓ Cette 3ème version (datant de 1995) est la plus flexible grâce au champ **d'extensions** qui peuvent être définies, par une communauté d'utilisateurs, dans le but de contenir des informations qui lui sont spécifiques et utiles.
    - ✓ Chaque extension est caractérisée par trois informations :
      - l'identifiant de l'extension considérée,
      - le fait qu'elle soit critique ou non,
      - la valeur de l'extension, propre à un certificat.
    - ✓ Le champ d'extension(s) peut ne contenir aucune extension.

Certificat numérique :

- **Certificat X.509 V3 (2/3):**



CERTIFICAT

Contenu du certificat (Données certifiées)

Version 3
Numéro de série du certificat
Informations sur la signature du certificat par l'AC (algorithmes et paramètres)
Nom du fournisseur du certificat
Période de validité du certificat
Nom du porteur de certificat
Informations sur la clé publique (valeur de la clé publique, algorithme et paramètres)

Extensions du Certificat

Identifiant du type de l'extension	Criticité (oui / non)	Valeur
Identifiant du type de l'extension	Criticité (oui / non)	Valeur
Identifiant du type de l'extension	Criticité (oui / non)	Valeur
…		

Algorithme de signature du certificat par l'AC

Algorithmes
Paramètres

Signature numérique du contenu du certificat

Valeur de la signature numérique du certificat par l'AC

Certificat numérique :

- **Certificat X.509 V3 (3/3):**
  - Des extensions sont standardisées , comme :
    - ✓ SubjectAlternativeName :
    - ✓ IssuerAlternativeName :
    - ✓ KeyUsage : usage de la clé certifiée (signature, chiffrement de clé, signature de certificat, …)
    - ✓ CertificatePolicies : indique les **politiques de certificat** supportées par l'émetteur;
    - ✓ PolicyMappings : indique l''équivalence' entre politiques de domaines différents;
    - ✓ PolicyConstraints :permet à l'émetteur d'appliquer des contraintes concernant les politiques de certificats sur les autres émetteurs dans un chemin de certification.;
    - ✓ Etc.

Certificat numérique :

- **Autorités de certification :**
  - Les émetteurs de certificats sont –à priori- des tierces parties de confiance qui ont donc pour rôle d'émettre des certificats qui permettent à des entités communicantes de s'authentifier en prouvant l'authenticité de leurs clés publiques et donc de leur signature.
  - Ces émetteurs sont appelés des autorités de certification ou 'CAs'.
  - Une **CA** est généralement un organisme, publique ou privé, qui jouit d'une notoriété au sein d'une communauté (qui peut être restreinte ou large) de telle sorte que cette dernière croit à l'authenticité des certificats qu'elle émis.

Certificat numérique :

- **Politique de certificat :**
  - Dans X.509, elle est définie comme étant « Un ensemble nommé de règles qui indiquent l'applicabilité d'un certificat à une communauté particulière et/ou à une classe d'applications avec des exigences de sécurité communes ». Elle détermine :
    - ✓ le niveau d'assurance attribué au certificat ;
    - ✓ le mode de vérification de l'identité des titulaires ;
    - ✓ la durée de validité des certificats ;
    - ✓ La révocation (utilisation des CRL)
    - ✓ Les limites de responsabilité
    - ✓ Le niveau des contrôle de sécurité et des audits, etc.

Certificat numérique :

- **Politique de certificat (2/2):**
  - Une CA pour montrer qu'elle est conforme à une politique de certificat, elle doit publier un CPS (Certificate Practice Statement) dont le rôle est de décrire de manière plus détaillée les pratiques d'émission et de gestion de certificats suivies par cette CA.
  - Le CPS représente donc la manière dont la politique de certificat est appliquée.
  - La confiance 'dans' un certificat émis par une CA dépend de sa politique de certificat et de son CPS.

II.3. Moyens de sécurité actuels

Certificat numérique :

- **Types de certificat :**
  - Il existe principalement deux types de certificats :
    - ✓ Le certificat d'identité où la partie réservée à l'information concernant le propriétaire de la clé publique permet de l'identifier.
    - ✓ Le certificat d'autorisation ou d'attributs où on ne s'intéresse pas à l'identité de l'entité certifiée mais à un certain nombre de ses attributs (tranche d'âge, profession, possession d'une licence spéciale, etc.).
  - Ces deux types de certificats diffèrent aussi en termes de :
    - ✓ durée de vie (généralement plus courte pour un certificat d'attributs)
    - ✓ Portée d'application (généralement plus réduite pour un certificat d'attributs)
    - ✓ Autorité émettrice (AA vs CA)

Une CA ou une PKI?

- Une CA est un composant d'une PKI parmi d'autres
- Une PKI peut contenir plusieurs CAs
- L'importance d'autres types d'entités que les CAs
- Une PKI est donc plus qu'une simple CA

III. Infrastructures de clés publiques PKI

Sommaire

III.1. PKI : Définitions, Rôle & Fonctionnement

➢ **Définition 1**

Un ensemble d'entités, communiquant par des protocoles et offrant des services pour la gestion (création, distribution, révocation, …) des clés publiques et de leur certificat (≈ IETF).

➢ **Définition 2**

Un système global d'authentification, de gestion des relations de confiance et de méthodes de protection de la confidentialité où les autorités de certification (CA) agissent comme des émetteurs de certificats

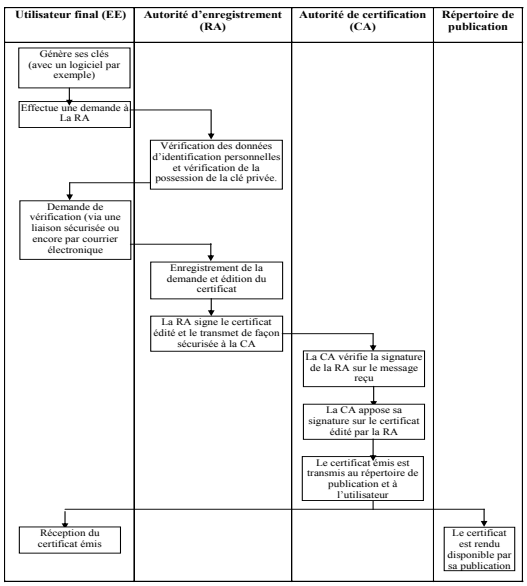
➢ Une PKI est généralement composée de plusieurs CAs afin de permettre l'utilisation des certificats dans un contexte large où une seule CA ne peut assurer à elle seule la gestion et la distribution de tous les certificats.

III.1. PKI : Définitions, Rôle & Fonctionnement

- **Entités** :
  - Entité finale (EE) : le titulaire d'un certificat et le vérificateur d'un certificat,
  - Autorité d'enregistrement(RA),
  - Autorité de certifications(CA)
  - Autorité de certification de politique (PCA et/ou PAA).
- **Méthode de révocation/validation** : liste de certificats révoqués (CRL) ou validation en ligne (exp. On-line Certificate Status Protocol :OCSP)
- **Politique de certificats** : un ensemble nommé de règles qui indiquent l'applicabilité d'un certificat à une communauté particulière et/ou à une classe d'applications avec des exigences de sécurité communes .
- **Modèle de confiance** : architecture, relations de confiance, etc.
- **Les protocoles de gestion de certificats**

III.1. PKI : Définitions, Rôle & Fonctionnement

- **Emission d'un certificat** :



Rôle d'une PKI :

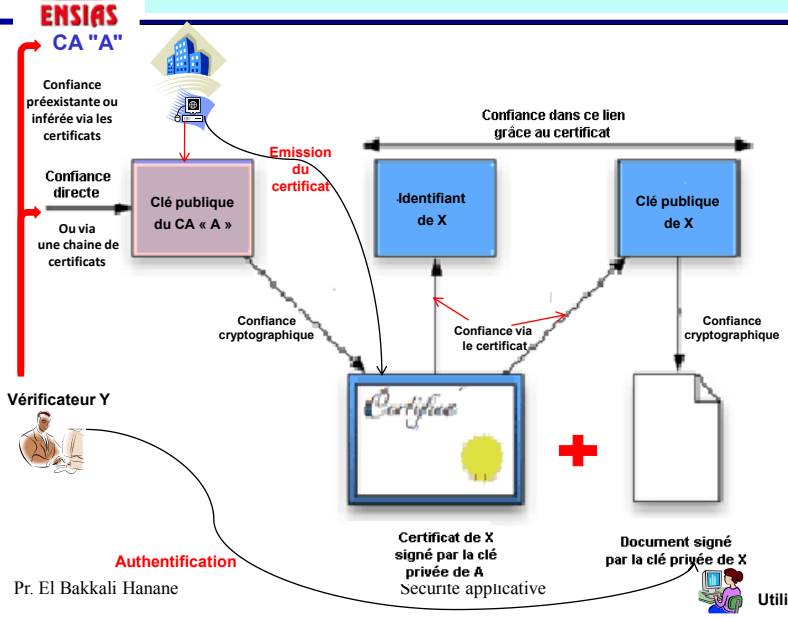
- Fournir les mécanismes nécessaires à établir des relations de confiance entre ses utilisateurs et leur offrir des services de sécurité tels que la confidentialité, l'intégrité, l'authentification et la non-répudiation et ce, essentiellement par le biais des certificats de clés publiques.
- Parmi les contextes où une PKI est très utile si ce n'est nécessaire, on trouve :
  - L'e-commerce ;
  - L'e-gouvernement ;
  - Le cas d'une grande entreprise possédant plusieurs applications qui nécessitent des services d'authentification et d'autorisation.

Confiance dans le 'contexte PKI'?

- Les relations de confiance entre entités d'une PKI se basent sur des relations préalables entre elles qui sont de deux sortes :
  - relations préexistantes en dehors de la PKI ;
  - relations nécessaires à l'adhésion à la PKI.
- Elles se basent aussi sur la confiance dans les techniques cryptographiques (algorithmes de cryptage & fonctions de hashage)
- La confiance n'est jamais totale ou absolue, elle est plutôt limitée par des contraintes et relative à un contexte donné.

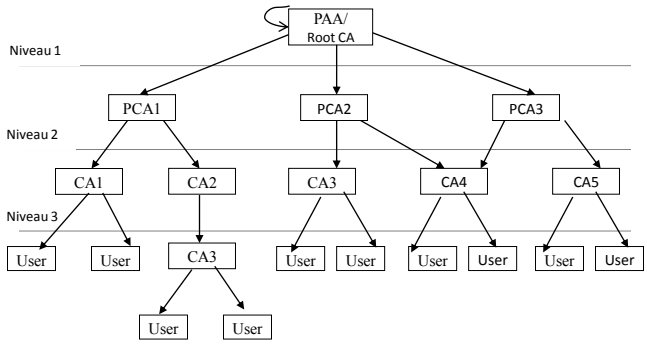
Confiance 'inférée' suite à l'émission d'un certificat

- Le vérificateur du certificat, dépendamment de sa confiance dans la CA émettrice et de sa politique de certification, peut 'inférer' les croyances suivantes :
  - La CA émettrice du certificat a été '*convaincue*' –suite à des procédures qui lui sont spécifiques - de l'authenticité du lien entre la clé certifiée et le sujet du certificat au moment de son émission. Ce lien peut concerner :
    - ✓ l'identité du sujet (certificat d'identité);
    - ✓ des attributs du sujets (certificat d'attributs ou d'autorisation).
  - Il est confiant - à un certain degré - de l'authenticité du dit lien.
- Confiance dans une CA : Compétence vs Honnêteté.



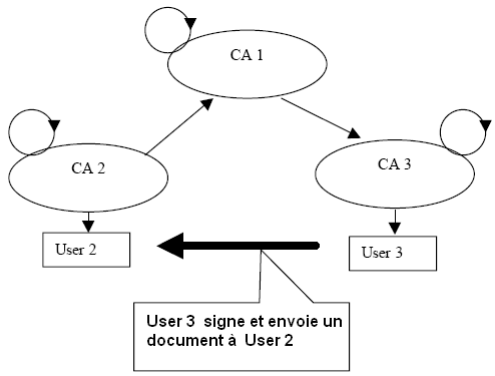
III.3. Modèle de confiance d'une PKI

➢ **Exemple d'un modèle hiérarchique** :



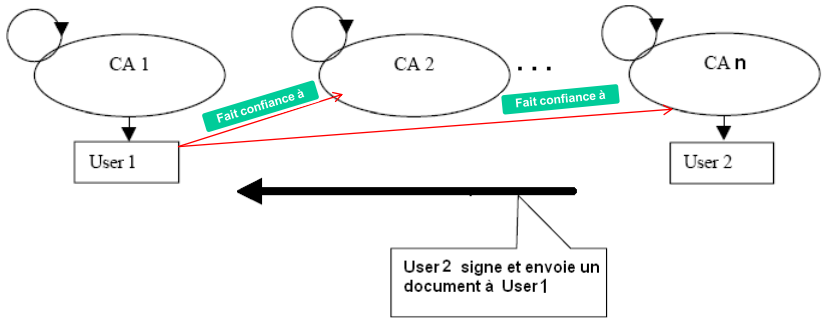
III.3. Modèle de confiance d'une PKI

➢ **Exemple d'un modèle 'réseau' avec des cross-certifications** :



III.3. Modèle de confiance d'une PKI

➢ **Exemple du modèle 'Liste de confiance'** :



III.4. Types d'une PKI

➢ PKI globales vs fermées :

- Globales : PKI préconisée par les normes X500/X509 ou la PKI de PGP (web of trust).
- 'Fermées' ou 'autonomes' d'entreprise :
  - ✓ En général, une seule autorité de certification suffit.
  - ✓ Achat de solution 'clé en main' parfois open source (exp : NewPKI, IDX-PKI de IDEALX, …)
  - ✓ Marché de plus en plus florissant des vendeurs de solutions PKI.

➢ PKI gouvernementales vs privées :

- Gouvernementales: PKI du Canada, PKI fédérale des U.S.A, L'IGC/A (Infrastructure de Gestion de la Confiance de l'Administration) de la DCSSI française , CertEurope, La Poste marocaine, …
- Privées (prestataires de service de certification): Verisign devenu External Certificate Authority en 2005 (agréé par les agences fédérales des U.S.A), Entrust, KEYNECTIS, etc.

III.5. Limitations d'une PKI

- Problème de déploiement et de gestion
- Problème d'interopérabilité
- Coût élevé
- Non comprise ou non appréciée par les utilisateurs
- Politiques de certificats très 'légères' ou ignorées
- Problèmes techniques (révocation, protection des clés privés, veille technologique, …)
- PKI globale difficile à atteindre
- Complexe et fait appel généralement à d'autres infrastructures complémentaires (Horodatage, archivage, notarisation, …)
- etc.

III.6. PKI : Conclusion

- La confiance dans la sécurité offerte par les solutions PKI ne peut être vraiment légitime que si les problèmes –techniques, organisationnelles et juridiques- posés par les technologies cryptographiques et autres moyens qu'elles utilisent soient résolues de manière satisfaisante.
- Satisfaisantes dans le sens où l'objectif visé n'est pas une sécurité absolue –qui n'est qu'une chimère- mais une sécurité comparable à celle offerte dans le monde conventionnelle.
- Le challenge concerne plutôt les PKI globales et beaucoup moins les PKI fermées
- Mais, les applications de type e/m- commerce ou e/m-gov ont plutôt besoin d'une large PKI.
- Tous les gouvernements du monde y travaillent dans ce sens.

Sommaire du cours

IV. Infrastructures complémentaires

Autorité d'horodatage :

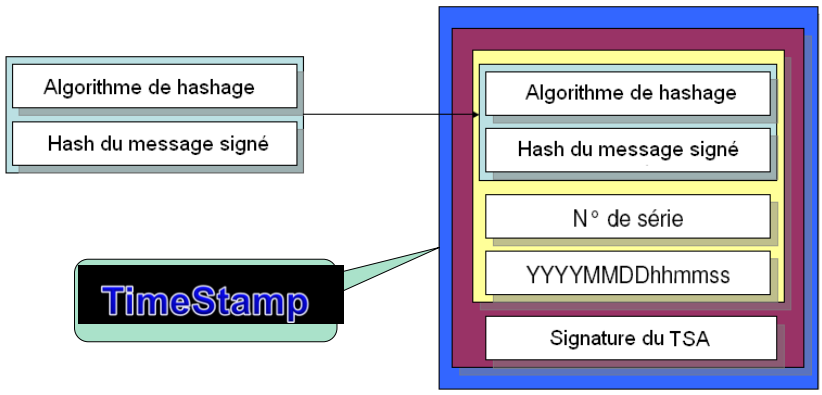
- **Définition :**
  - Une Autorité d'Horodatage ou Time Stamp Authority (TSA) peut être définie comme une tierce partie de confiance dont le rôle est de certifier des heures et des dates.
  - Une TSA est généralement associée à une PKI et permet de renforcer les services de sécurité offerts par la PKI (surtout la non-répudiation).
- **Buts de l'horodatage :**
  - Prolonger la durée de vie d'une preuve signée au-delà de la durée de vie de la signature (ou du certificat correspondant) en apportant la preuve d'antériorité ;
  - Éviter le renvoi à une date ultérieure d'un message par une entité qui a pu se le procurer (protection anti-rejeu ou replay).

Autorité d'horodatage:

- **Principe :**
  - **Compléter** la signature électronique d'un document par la date de signature ➔ signature électronique de la date connue sous le nom de contremarque ou jeton de temps (Time Stamp).
  - Un horodatage sécurisé doit se baser sur une source de temps sécurisée et reconnue fiable.
  - Si les certificats sont utilisés (ce qui est souvent le cas), il faut prolonger la durée de conservation des certificats et des listes de certificats révoqués (CRL).
  - La RFC 3161 définit le format des Time Stamps et le protocole TSP (Time Stamp Protocol).

Autorité d'horodatage:

- **Jeton d'horodatage (TimeStamp):**



Autorité d'horodatage:

- **Services offerts :**
  - Garantie de date de signature,
  - Preuves de possession,
  - Preuves d'existence et d'antériorité,
  - **Renforcement de la non-répudiation,**
  - Garantie de l'heure d'une transaction (commerciale ou administrative),
  - Garantie des heures et des dates des listes de certificats révoqués (CRL),
  - Messages avec accusés de réception,
  - **Notarisation et archivage sécurisés**,
  - ...

Autorité d'horodatage:

- **Quelques questions importantes :**
  - Valeur légale du TimeStamp émis par une TSA ?
  - Durée de validité du TimeStamp (5 ans, 30 ans, …) ?
  - Validité du TimeStamp dans le cas où la TSA a arrêté ses activités ?
  - Exigences pour une TSA de confiance ?
  - Combinaison d'une signature non certifiée (non présumée fiable) et un TimeStamp qualifié ?
  - Taille de clé de l'algorithme de cryptage ou la sécurité de la fonction de hashage ne sont plus suffisants ?

Archivage sécurisé:

- **But :**
  - Conserver « en l'état » les contenus qui lui sont confiés tout en assurant de les restituer à l'identique ultérieurement.
- **Utilisation avec une PKI :**
  - L'archivage sécurisé des document signés, des certificats et toutes informations utiles relatives à la certification vise à se préparer aux éventuels besoins de restitution (par exemple, suite à des litiges) qui peuvent avoir lieu après l'expiration ou la révocation d'un certificat ou l'arrêt de service d'une CA.
  - L'archivage peut être interne ou via un tiers archiveur externe et indépendant jouant le rôle d'une **autorité d'archivage** de confiance (TAA: Trusted Archive Authority).

Autorité d'archivage:

- **Services offerts :**
  - Sauvegarde sécurisée (intégrité préservée) des documents électroniques qui lui sont confiés (exp, certificats, CRL, politiques de certificats, …) ;
  - Conservation active **sur le long terme** en prenant en charge **l'évolution des supports de stockage** (changement des mediums de sauvegarde lorsque cela s'impose) et des technologies cryptographiques ;
  - Mise à disposition des documents archivés aux personnes **autorisées** ou administrations mandatées ;

Sommaire du cours

V. Sécurité du e-commerce

Définitions :

- L'ensemble des **échanges électroniques** liés aux activités commerciales : flux d'information et transactions concernant **la vente et l'achat de biens ou de services**;
- L'ensemble des **échanges commerciaux** dans lesquels l'achat s'effectue sur un **réseau de télécommunication** (surtout **Internet**) ;
- On peut inclure dans le commerce électronique l'ensemble des **usages commerciaux des réseaux** (exp : simple catalogue sur Internet).
- On parle donc du e-commerce même si seule la commande est 'électronique' contrairement au paiement et à la livraison qui sont effectués par des méthodes traditionnelles.

Types du e-commerce :

- **B2C :**
  - Business to consumers = entre un commerçant (ou une entreprise) et un consommateur.
- **B2B :**
  - Business to business = commerce interentreprises (exp: commerce entre une entreprise et son fournisseur).
- **B2G :**
  - Business to government = entre une entreprise et une Administration.
- **C2C :**
  - Consumer to Consumer = entre deux particulier (une sorte de troc).

Succès du e-commerce

- **Auprès des entreprises, Il s'explique par différents facteurs, comme :**
  - La présence sur Internet contribue significativement au succès commercial d'une entreprise (accès à de nouveaux marchés, de nouveaux clients, …) ;
  - Les TIC permettent une plus grande souplesse d'adaptation au consommateur et le développement d'un **marketing personnalisé** → augmentation de la demande ;
  - La diversification de l'offre et la **création de nouveaux services** devient plus simple ;

Succès du e-commerce

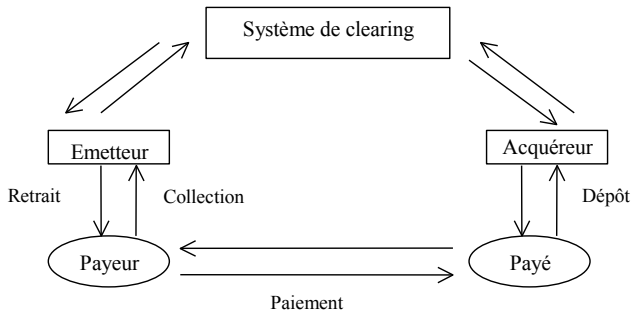
- **Auprès des consommateurs , Il s'explique par :**
  - Démocratisation d'Internet ;
  - De nombreux avantages spécifiques à l'e-commerce (absence de contraintes de temps ou d'espace, comparaison facilitée entre les produits et les prix, nouveaux services, ventes aux enchères, …) ;
  - Une meilleure confiance grâce au renforcement de la sécurité (notamment du paiement électronique) et du cadre réglementaire et aussi grâce à l'ancienneté des sites Web marchands ;
  - L'avènement du m-commerce (encore plus convivial)
  - ….

Description

- L'e-paiement fait relier les mêmes acteurs que le paiement conventionnel, on a donc au moins un acheteur (payeur) et un vendeur (payé).
- Des intermédiaires financiers comme les banques, les opérateurs de cartes, les systèmes de change et de compensation peuvent intervenir selon le moyen de paiement utilisé.
- L'infrastructure informatisée reliant ces intermédiaires financiers est déjà en place à l'échelle mondiale. L'apport du e-paiement réside dans **l'informatisation de la relation entre le payé, le payeur et l'univers financier**, en général.

Architecture

- La figure suivante montre l'architecture générale d'un système de paiement électronique avec les différentes transactions entre ses intervenants :



Intervenants

- L'émetteur est un organisme (en général, la banque du payeur) qui émet au payeur un instrument de paiement électronique valide.
- L'acquéreur est un organisme (en général, la banque du payé) que le payé a chargé de vérifier la validité de l'instrument de paiement utilisé par le payeur lors de la transaction de paiement et de créditer ensuite son compte du montant de la transaction.
- Le payeur retire des instruments de paiement de l'émetteur, pour les utiliser ultérieurement dans des transactions de e-paiement.
- Le payé, en général un commerçant, en échange des biens ou services qu'il fournit au payeur, reçoit de ce dernier, une donnée (on parle de 'transcript') spécifique à l'instrument de paiement utilisé et qui est 'convertible en argent' via son acquéreur.

Méthodes du e-paiement

➢ **Instruments de paiement :**

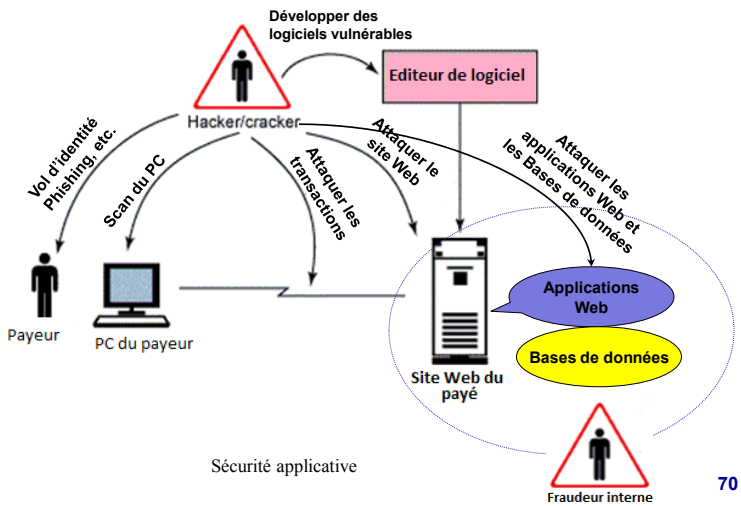
- A l'heure actuelle, nous distinguons trois types d'instruments de paiement électronique qui sont tous inspirés d'un moyen de paiement conventionnel :
  - ✓ Le chèque électronique ou e-chèque : problème d'extensibilité ;
  - ✓ La monnaie électronique ou e-cash : utilisée surtout pour les micro-paiements avec possibilité d'anonymat;
  - ✓ La carte de crédit (ou de débit) : le moyen le plus utilisé de par le monde (plus de 80%) pou les paiements en ligne.

Méthodes du e-paiement

- **Paiement par carte de crédit :**
  - Obligations pour le commerçant (et ses prestataires de service) :
    - ✓ Conformité au standard **PCI DSS** (Payment Card Industry Data Security Standard) qui spécifie les exigences de sécurité à respecter, en particulier, en termes de protection des données des cartes de crédits (applicable depuis 2005, la dernière version V3.0 –plus exigeante- applicable depuis juillet 2015).
    - ✓ Conformité aux lois de sa juridiction

Risques de Sécurité

- **Les risques se situent à différents niveaux :**



Sécurité des transactions

- **Le paiement électronique est la transaction la plus critique dans le déploiement du commerce électronique, il requiert généralement :**
  - **L'authentification** de toutes les parties impliquées dans la transaction de paiement (dans une moindre mesure le payeur) ;
  - **L'intégrité** des données échangées ;
  - La **confidentialité** de ces données ou du moins celles à caractère financier ou personnel ;
  - La **non-répudiation** des transactions (par le payeur ou le payé) ;
  - **L'anonymat** du payeur peut être un besoin de sécurité ;
  - La **convivialité** (et un temps de réponse réduit) ; pourra
  - La **traçabilité** des transactions

Un protocole sécuritaire adéquat

Sécurité des transactions

- **Les opérateurs de cartes de crédit jouent un rôle important dans la sécurisation de la transaction de paiement électronique :**
  - Participation à l'élaboration de protocoles sécuritaires spécialisés (SET, 3D-Secure)
  - Spécification du standard PCI DSS pour la protection des données de cartes de crédit;
  - En cas de fraude, les payeurs victimes disposent de 120 jours pour contester la transaction et dans ce cas, c'est généralement le commerçant qui a reçu le paiement qui supporte le remboursement.

Sécurité du site Web

- **Elle requiert généralement :**
  - La sécurisation du serveur hébergeant le site ;
  - La protection des données personnelles stockées sur le site du payé ;
  - Une haute disponibilité (redondance) ;
  - Une bonne résistance aux attaques de type :
    - ✓ Attaques DNS
    - ✓ Injection de code malveillant
    - ✓ Ajout de liens cachés
    - ✓ …

Sécurité des applications Web

- **Elle requiert généralement :**
  - L'intégration des besoins de sécurité depuis la phase conception ;
  - Un développement sécurisé (code résistant aux attaques) ;
  - Une base de données sécurisée (avec un contrôle d'accès bien défini et bien appliqué);
  - Une bonne stratégie de sauvegarde (qui est appliquée convenablement);
  - Une 'bonne' authentification des utilisateurs ;
  - Un 'bon' filtrage applicatif , …

Une bonne résistance aux attaques

Historique

- **Protocole SSL V2.0 (1994)**
  - Domaine d'application plus large que l'e-commerce
- **Protocole SSL V3.0/ TLS 1.0 (1999)**
  - Utilisés surtout pour sécuriser HTTP
- **Protocole SET (1997- (≈)2002)**
  - Sécurisation des paiements par Cartes de crédit
- **Protocole 3D-Secure (2002)**
  - Authentification des porteurs de cartes de crédit (3DS 2.0 attendue en principe en 2016)

Protocoles SSL /TLS

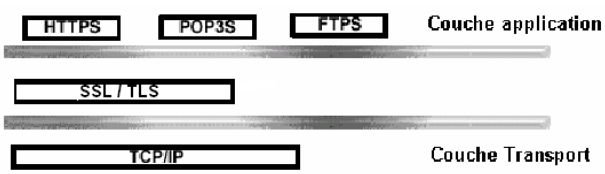
- **SSL ou TLS ?**
  - Le protocole SSL (Secure Socket Layer) a été conçu en 1994 par Netscape mais, la version 3 a reçu les contributions de la communauté internationale (IETF en 1999). Depuis, l'IETF l'a renommé en TLS (Transport Layer Security) et sa documentation se trouve dans la RFC 2246 .
  - La version 3 de SSL et TLS V1.0 ont très peu de différences et la plupart des serveurs et des navigateurs web peuvent mettre en œuvre les deux protocoles.
  - De ce fait, dans la littérature on parle de SSL/TLS.

Protocoles SSL /TLS

- **Principales différences TLS vs SSL**
  - Plus de messages d'erreurs
  - Amélioration des calculs crypto
  - Enrichissement de la CipherSuite (exp : SHA-256)
  - Possibilité de gérer sur le port 80 aussi bien HTTP que HTTPs

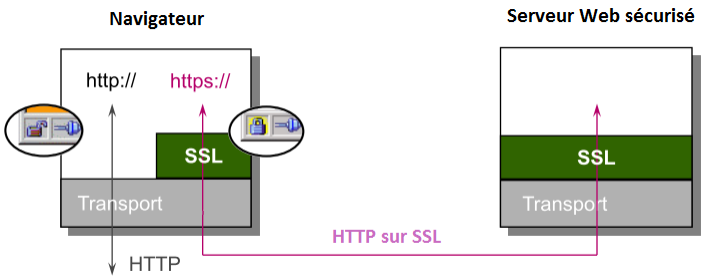
Protocoles SSL /TLS

- **Description :**
  - Il se situe juste au dessus de la couche transport (couche session) afin d´assurer la sécurité des communications de deux applications client/serveur :



Protocoles SSL /TLS

- **Utilisation courante :**
  - Il est essentiellement utilisé pour sécuriser la connexion entre un navigateur et un site Web :



Protocoles SSL /TLS

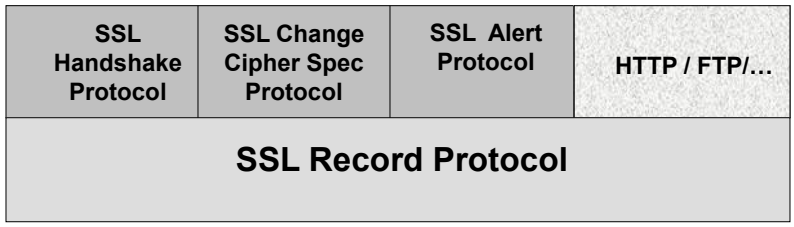
- **Objectifs :**
  - La sécurité : il permet d'établir une connexion sécurisée en termes de confidentialité, d´authentification et d'intégrité ;
  - L'interopérabilité : les applications utilisant SSL/TLS peuvent échanger les paramètres cryptographiques sans connaissance d'autres codes ;
  - L'extensibilité : il offre un cadre dans lequel de nouvelles méthodes de cryptage peuvent être incorporées (sans besoin d'un nouveau protocole) ;
  - L'efficacité :
  - il utilise le cryptage symétrique pour le cryptage de masse et celui à clé publique pour l'échange de clés de session;
  - Une session SSL peut inclure des connexions sécurisées multiples.

Protocoles SSL /TLS

- **Principe :**
  - Client et serveur commencent par une phase d'authentification (qui est optionnelle pour le client) ;
  - Ils négocient ensuite une clé symétrique de session qui servira à assurer la confidentialité des échanges ;
  - L'intégrité de ces échanges est assurée par l'application d'un MAC (Message Authentification Code) à chaque message. Pour plus d'efficacité, ce sont des HMAC (Hashed Message Authentification Code) qui sont utilisés.
  - A la fin des échanges, le client ou le serveur initient la fermeture de la session.

Protocoles SSL /TLS

- **Architecture :**
  - SSL est un protocole à 2 couches. Pour chacune d'elles, les messages incluent des champs de longueur, de description et de contenu :



Protocoles SSL /TLS

- **État de session :**
  - La négociation de l'état de session entre le client et le serveur se fait lors du 'Handshake' et il comprend plusieurs éléments, dont :
    - ✓ Session identifier: un octet fixé par le serveur pour identifier la session ;
    - ✓ Peer certificate : un certificat pour le serveur, éventuellement un autre pour le client;
    - ✓ Compression method (par défaut Null) ;
    - ✓ Cipher spec : spécifie les algorithmes de cryptage et de calcul de MAC;
    - ✓ Master secret : secret partagé (48 octets) entre le client et le serveur ;
    - ✓ Is resumable : flag qui indique si de nouvelles connexions peuvent être créées à partir de cette session.

Protocoles SSL /TLS

- **État de connexion:**
  - L'état d'une connexion comprend également plusieurs éléments, dont :
    - ✓ Server and client random ;
    - ✓ Server write MAC secret : le secret utilisé dans le calcul du MAC par le serveur ;
    - ✓ Client write MAC secret ;
    - ✓ Server write key : la clé symétrique de cryptage pour les données cryptées par le serveur et décryptées par le client ;
    - ✓ Client write key ;
    - ✓ Sequence Number : chaque partie maintient des n° de séquence séparés pour les messages transmis et reçus.
    - ✓ Initialization vectors : pour un algorithme de chiffrement par bloc en mode CBC. Le premier est fixé lors du handshake, les suivants sont les derniers blocs des précédents fragments chiffrés

Protocoles SSL /TLS

- **SSL Record Layer** :
  - Lorsque cette couche reçoit un 'message' Mesg de taille arbitraire d'un protocole de la couche supérieure, elle procède comme suit :
    - ✓ Fragmentation : Mesg → n fragments dits SSLPlaintext records (taille maximale =2 14 octets) ;
    - ✓ Compression (prévue mais non supportée): SSLPlaintext record → SSLCompressed selon l'algorithme spécifié dans l'état de session courant ;
    - ✓ Protection (ajout du MAC et cryptage) : SSLCompressed → SSLCiphertext selon les algorithmes de cryptage et de calcul de MAC spécifiés dans le CipherSpec de l'état de session courant ;
    - ✓ Encapsulation SSL : ajout d'entête (type, n° de version et longueur).

Protocoles SSL /TLS

- **SSL Record Layer** :
  - Les opérations inverses (de celles pré-citées) sont effectuées par ce protocole sur les données destinées aux couches supérieures :
    - ✓ Décryptage et vérification du MAC (confidentialité et intégrité) ;
    - ✓ Décompression (si la compression a eu lieu) ;
    - ✓ Réassemblage des fragments.

Protocoles SSL /TLS

- **SSL Change Cipher Spec :**
  - Ce protocole a pour but de signaler les transitions dans les stratégies de cryptage et de calcul de MAC.
  - Il consiste en un seul message codé sur un octet de valeur 1.
  - Le serveur (resp. client) doit envoyer ce message au client (resp. serveur) pour l'aviser du changement des clés et des algorithmes utilisés par SSL Record Layer pour la protection des données

Changement du CipherSpec (resp. Compression method) courant de la session par celui en instance (qui a été négocié dans le HandShake).

Protocoles SSL /TLS

- **SSL Alert :**
  - Ce protocole permet l'échange entre client et serveur d'un certain nombre de messages d'alerte (de deux octets) qui peuvent être de deux types :
    - ✓ Alerte de clôture : chaque partie peut initier la fermeture d'une connexion par l'envoie d'un message Close_notify à la partie homologue, celle-ci doit répondre par un autre message Close_notify et fermer immédiatement la connexion en abandonnant toutes écritures en instance.
    - ✓ Alertes d'erreurs : quand une erreur est décelée par une partie, celle-ci envoie un message d'alerte à l'autre partie. Si l'erreur est 'fatale' les deux parties ferment immédiatement la connexion. Des exemples d'alerte d'erreurs sont:
      - Bad_record_mac : si le message reçu a un MAC erroné, cette alerte est fatale;
      - Handshake_failure : Si l'émetteur n'a pas pu négocier des paramètres de sécurité acceptable avec le récepteur, cette alerte est aussi fatale.

Protocoles SSL /TLS

- **SSL Handshake :**
  - Ce protocole permet au client et au serveur de s'authentifier, de négocier une 'Cipher Suite' comprenant les algorithmes de cryptage et de MAC et enfin d'échanger les clés de session (symétriques).
  - Plusieurs algorithmes symétriques sont autorisés par SSL dont : AES, 3-DES 168, IDEA 128, RC4 128, …
  - Les fonctions de hashage sont MD5 et SHA-1.
  - L'algorithme à clé publique préféré est RSA et le format de certificat X.509 V3.
  - DH peut être utilisé pour l'échange de clés à la place de RSA.

Protocoles SSL /TLS

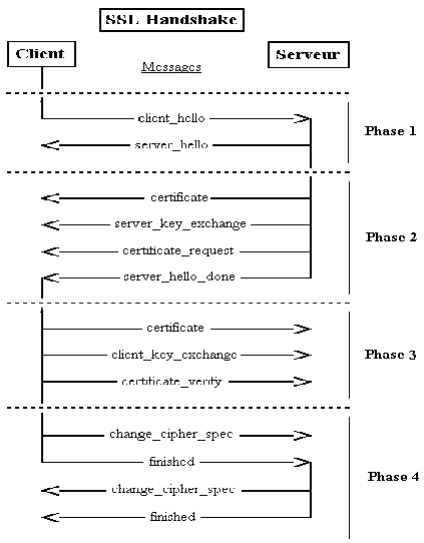
- **SSL Handshake :**

✓ **Phase 1 :** Négociation des paramètres de sécurité entre client et serveur (Version SSL, algorithmes d'échange de clé et de cryptage, etc.) via deux messages 'Hello'.

✓ **Phase 2 :** Authentification du serveur (envoi du certificat) et échange des clés.

✓ **Phase 3 :** Authentification (ou message d'alerte No-certificate) du client et échange des clés.

✓ **Phase 4 :** Envoi du message *Change Cipher Spec* et fin du Handshake (par échange du message 'finished').



Protocoles SSL /TLS

- **Limitations :**
  - SSL assure seulement la sécurité des communications (point à point) et non la sécurité au niveau utilisateur final;
  - La non-répudiation n'est pas assuré actuellement par SSL (pas de signature de données), ce qui est un point faible pour son utilisation dans l'e-paiement;
  - La liste des CA de confiance (distribuée dans les navigateurs) utilisés par SSL ne bénéficie pas vraiment de la confiance des utilisateurs (leur rôle est quasi-transparent);
  - La consultation des CRL est quasi-absente;
  - SSL n'offre pas une sécurité élémentaire ou sélective (une partie d'un message, par exemple) puisque il sécurise l'ensemble des données échangées (cela est dû qu'il est indépendant des applications du niveau supérieur), etc.

Protocole SET

- **Description:**
  - En janvier 96, Visa et MasterCard ont annoncé avec d'autres partenaires, le développement d'un ensemble de spécifications techniques permettant la sécurité du paiement par carte de crédit sur les réseaux ouverts.
  - Ce Standard a été appelé SET (Secure Electronic Transaction), il a été rendu publique dès février 1997, et de nombreux vendeurs ont alors commencé à développer des applications d'e-commerce basées sur SET.
  - SET assure la sécurisation des dialogues entre le client et le commerçant d'une part, et entre ce dernier et une entité qui joue le rôle de passerelle de paiement (payment gateway) d'autre part.

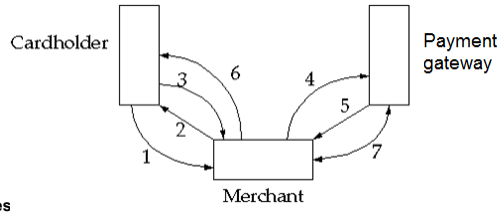
Protocole SET

- **Principe :**
  - Le rôle de la passerelle de paiement peut être joué par l'acquéreur (e.g la banque du commerçant) qui peut également assurer le rôle d'une autorité de certification pour le commerçant.
  - SET consiste en des paires de messages requête/réponse. Le cryptage est appliqué différemment à certaines parties des messages, afin de permettre à l'information contenue dans le message d'être **sélectivement** révélée aux différentes parties :
    - ✓ Les données financières concernant la carte de crédit ne sont pas révélées au commerçant ;Celles concernant le produit acheté ne sont pas montrées à l'acquéreur.
    - ✓ Un cryptage point à point (par exemple, avec SSL) n'aurait pas permis ce type de sélection.

Protocole SET

- **Requêtes/Réponses:**

1 : PInitReq
2 : PInitRes
3 : PReq
4 : AuthReq
5 : AuthRes
6 :PRes
7 :CapReq/CapRes

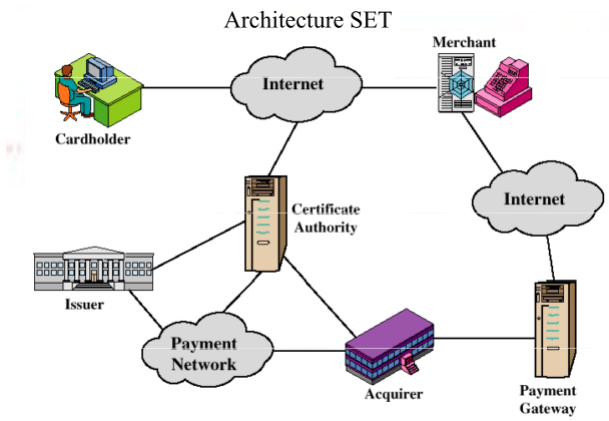


- La commande d'achat (Payment Request:PReq) est le message le plus complexe de ce protocole. Le payeur envoie deux éléments de message au commerçant: L'information de commande (Order Information:OI) et les instructions de paiement (Payment Instructions :PI).

Payment Request

Protocole SET

- **Architecture :**



Protocole SET

- **Signature duale:**
  - Elle est constituée de la manière suivante:
    - ✓ D'abord, le résultat H2 d'une fonction de hachage appliquée aux données résultantes de la concaténation du 'hash' des données de la phase d'initialisation (PInitReq/PInitRes) et d'un Nonce aléatoire: H(M1) et du 'hash' des données de paiement (PIdata) utilisées dans PI : H(M2), soit H2=H(H(M1),H(M2)), est signé par le client: {H2}sigC.
    - ✓ Ensuite, une opération XOR est effectuée entre H(M1) et H(M2) pour obtenir Y=H(M1) XOR H(M2).
    - ✓ Enfin, la signature duale n'est autre que ({H2}sigC, Y)

Protocole SET

- **Signature duale:**
  - Elle est vérifiée de la manière suivante :
    - ✓ Le commerçant connaissant M1 (lors de la phase initialisation) peut vérifier la signature en effectuant les étapes suivantes:le calcul de H(M1), l'extraction de H(M2) en calculant H(M1) XOR Y, le décryptage de {H2}sigC et enfin la comparaison de H2 avec H(H(M1),H(M2)). L'égalité prouve la validité de la signature.
    - ✓ De la même manière, l'acquéreur peut vérifier la signature, connaissant cette fois M2 (Les données financières de PI sont cryptées par la clé publique de l'acquéreur).
    - ✓ Cette manière de faire permet de limiter la vérification à une seule signature.

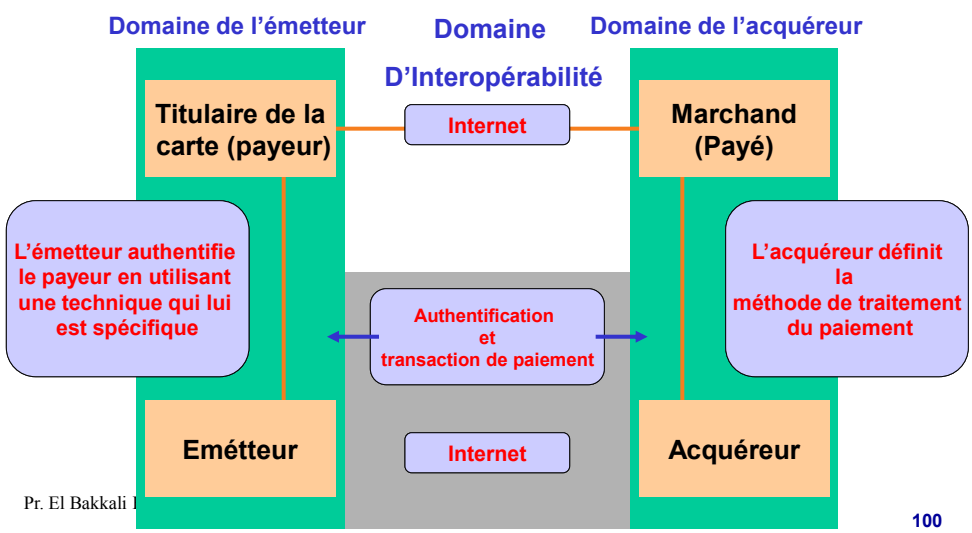
Protocole SET

- **Mots de la fin :**
  - Il n'a pas eu le succès attendu (il est pratiquement mort après presque 5 années d'existence):
    - ✓ il repose sur l'existence d'une PKI hiérarchique pour la gestion des certificats, pour l'authentification du payé, de la passerelle de paiement et du payeur qui est, actuellement, différente pour chaque solution de e-paiement basée sur SET ce qui empêche l'interopérabilité et l'extensibilité de ces solutions ;
    - ✓ Il implique un coût d'implémentation élevé pour les commerçants et une plus grande complexité de mise en œuvre pour les acheteurs.
  - Visa et MasterCard, en réponse à cet échec, ont opté pour un nouveau protocole "3D-Secure" ("Verified by Visa" pour Visa et "Secure Code" pour MasterCard) qui se base sur SSL/TLS pour la confidentialité et l'intégrité.

Protocole 3D Secure

- **Description:**
  - A l'heure actuelle, c'est 3D-Secure qui veut s'imposer –mais jusqu'à maintenant sans vrai succès- comme remplaçant de SET et comme une alternative plus sûre que l'utilisation du seul SSL/TLS.
  - Contrairement à SET, ce protocole n'exige pas que le titulaire de la carte ait un certificat, l'authentification repose plutôt sur une information personnelle ou un mot de passe partagé avec l'émetteur (Issuer).
  - Il vise à réduire le risque de Fraude en garantissant que le payeur est bien le titulaire légitime de la carte.
  - Il se base sur un modèle de paiement à 3 domaines (d'où le 3D) qui est combiné avec un changement au niveau des responsabilités :
    - ✓ En cas de fraude, c'est l'émetteur (et non plus le commerçant) qui assume les paiements frauduleux.

Protocole 3D Secure



Protocole 3D Secure

- **Quelques limitations:**
  - Le Marchand doit souscrire au programme 3D-Secure de sa banque
  - L'acheteur doit également s'inscrire via sa banque, qui va choisir de l'identifier par un mot de passe, sa date de naissance, un identifiant...
  - Authentification de l'acheteur passe par la saisie de cet identificateur via une URL de sa banque → Réticence de certains acheteurs
  - Les banques aussi peuvent être réticentes (car désormais c'est elles qui se chargent des remboursements en cas de Fraude)

Remarques finales

- Le cryptage du n° de carte de crédit (comme cela est possible avec SSL) est loin d'être suffisant pour lutter contre la fraude, car si cela permet de prévenir son interception, il ne permet pas d'éviter sa divulgation auprès du marchand (ou des attaquants du site du marchand);
- L'implication d'un tiers de confiance permet d'assurer le payeur quant à la crédibilité du marchand. Mais, la non divulgation du n° de carte au marchand demeure toujours plus rassurante → SET permet justement d'éviter ce problème.
- La diminution de la fraude visée par 3D Secure n'a pas su peser –pour le moment- en sa faveur vis-à-vis des acheteurs (il protège surtout les Marchands).
- Les utilisateurs de 3D-secure deviennent plus vulnérables aux attaques de type phishing.

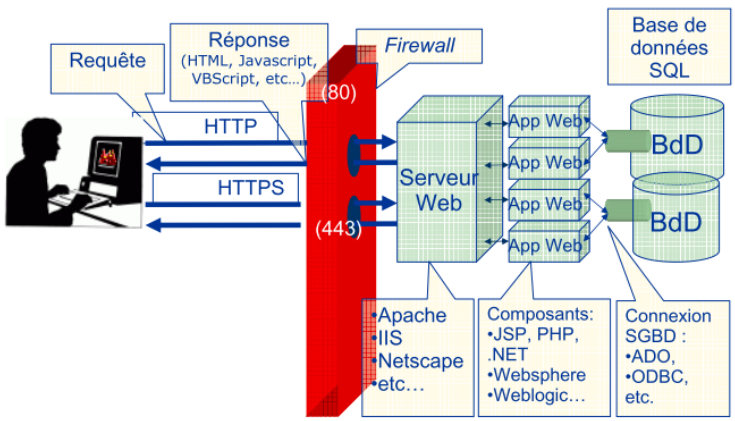
Sommaire du cours

VI. Sécurité des applications Web

Les applications Web dans l'entreprise

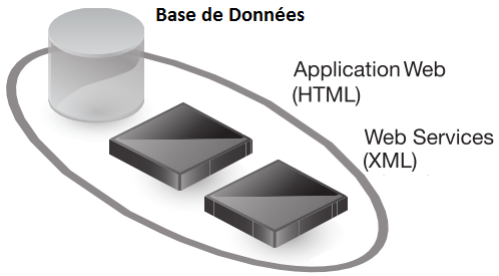
- Les applications Web sont de plus en plus ancrées dans l'activité d'une entreprise.
- Elles occupent souvent un rôle stratégique et représentent un élément critique et vital pour le fonctionnement quotidien de toute organisation moderne.
- Elles doivent donc répondre à des exigences en termes de disponibilité (souvent une haute disponibilité est requise) , de performance et de sécurité.
- Toute vulnérabilité pouvant affecter le fonctionnement normal des applications doit être anticipée et ce, dans la mesure du possible, dès la phase de conception.
- Il faut aussi pouvoir concilier **performance** & **sécurité** lors du déploiement.
- Des questions concernant par exemple, les ressources sensibles de l'application, les sources possibles des vulnérabilités potentielles ou la gestion de l'authentification et du contrôle d'accès, doivent être étudiées dans le cadre de la politique de sécurité applicative.

Architecture typique d'une application Web



Web services

- Les applications modernes sont souvent basées sur les Web Services dont l'une des caractéristiques est la communication entre applications (une sorte de "réseau applicatif ")
- Dans le cas où l'accès à une application est initié par une autre application (comme pour les architectures SOA), le contrôle d'accès devient encore plus complexe à mettre en oeuvre.



Sécurité d'une application Web

- **Une partie d'un tout :**
  - La sécurité d'une application Web ne peut être abordée indépendamment de la sécurité de l'OS du serveur, de la base de données (Back-end), du réseau, etc.
  - Une faille dans l'application peut permettre un accès non autorisé aux données, et ce même dans le cas d'une base de données cryptée.
  - Elle est donc plus que la sécurité du code de l'application ou de son déploiement.
  - La protection de l'utilisateur (légitime et honnête) en fait également partie.
- **Une partie très critique :**
  - La survie d'une entreprise peut dépendre de la sécurité de ses applications Web (Exemple du e-commerce).
  - Trop de sécurité peut faire fuir des clients !!

Sécurité d'une application Web

- **Approches et bonnes pratiques:**
  - OWASP **ASVS (**Application Security Verification Standard Project)
  - OWASP **SAMM** (Software Assurance Maturity Model)
  - OWASP Testing Guide
  - Norme **ISO/IEC 27034-2:2015 (Application security)**
  - **NIST 800-53 …**
  - **SANS ….**

Approches de la Sécurité des applications Web (source: SANS- Mai 2015)

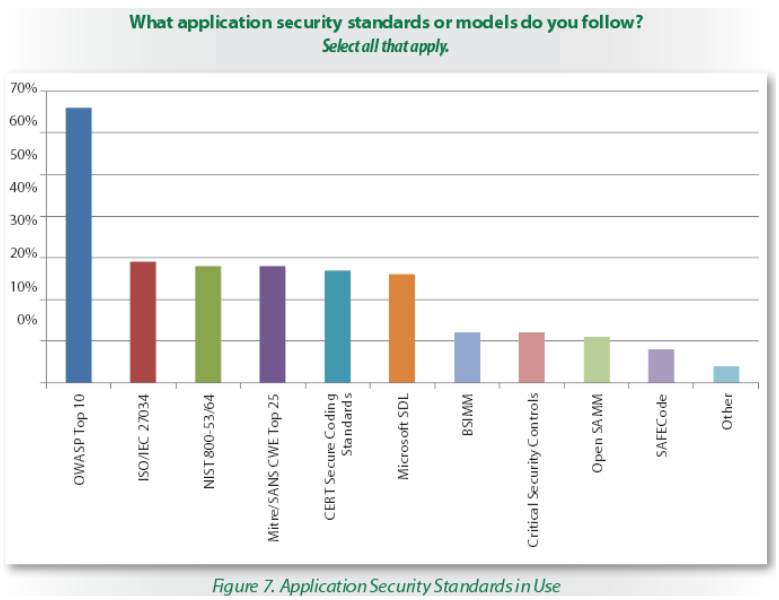


Figure 7. Application Security Standards in Use

VI. Sécurité des applications Web

1. Introduction
2. HTTP en bref
3. Principales attaques sur les applications Web
4. Firewall applicatif Web
5. Authentification des utilisateurs

VI.2. HTTP en bref

- **Aperçu général :**
  - HTTP (HyperText Transfer Protocol) : protocole de transfert de ressources **sans état** d'un serveur HTTP vers un client HTTP (navigateur).
  - HTTPv1.0: RFC1945(96) et HTTPv1.1: RFC2068 (97) & RFC2616 (99).
  - Il se base sur le protocole TCP (port 80) et les transferts des données se font en utilisant le format MIME.
- **Requêtes HTTP :**
  - Une requête HTTP utilise les méthodes GET, PUT, POST,etc :
  - GET: récupérer un document (exp : **GET** http://www.ec-lyon.fr/index.html HTTP/1.0)
  - HEAD: récupérer l'en-tête d'un document sans le contenu utile pour récupérer la date de dernière modification ou pour faire des tests de connectivité
  - POST: permet d 'envoyer des données au serveur (exp. formulaires)
  - PUT, DELETE: permettent d'ajouter des documents ou d'en supprimer

VI.2. HTTP en bref

- **Format des Requêtes HTTP :**
  - Une ligne pour la requête: METHODE URL-du-doc version
  - Plusieurs lignes d'en-tête (optionnelles):
    - ✓ En-tête générique/ En-tête de la requête/ En-tête de l'entité (en-tête MIME des données)
  - Une ligne vide séparatrice
  - Plusieurs lignes de données (pour POST et PUT)
  - Exemple:

```
POST http://www.ec-lyon.fr/cgi-bin/search  HTTP/1.0
Date: Mon, 30 sep 2003 10:00:50 GMT
User-Agent: Mozilla/4.03 [en] (Win98; I)
Content-Type: application/ x-www-form-urlencoded
Content-Length: 22

NOM=Bond&PRENOM=James
```


VI.2. HTTP en bref

- **Format des Réponses HTTP :**
  - Statut de la réponse (une ligne):
    - ✓ Version du protocole utilisé par le serveur (HTTP/1.0 ou HTTP/1.1) ;
    - ✓ Code numérique du statut (3 chiffres) et Statut de la réponse en anglais.
  - Plusieurs lignes d'en-tête : En-tête générique/ En-tête de la réponse/ En-tête de l'entité (en-tête MIME des données) .
  - Une ligne vide séparatrice
  - Plusieurs lignes de données
  - Exemple:

```
HTTP/1.0 200 document follows
Date: Mon, 30 sep 2003 10:00:50 GMT
Server: Apache/1.3.6 LV/LM-1.3 (Unix) PHP/3.0.9
Content-Type: text/html
Content-Length: 90
Last-Modified: Sun, 06 Nov 2002 08:49:37 GMT

<HTML> <HEAD> <TITLE>Premier essai</TITLE> </HEAD>
<BODY> Bonjour ! </BODY> </HTML>
```

Dialogue HTTP 1.0

- **Etapes de base :**
  - Le client (navigateur) 'demande' au DNS l'adresse IP du serveur 'nom-serveur'
  - Le client établit une connexion TCP sur le port 80 du serveur
  - Le client envoie la requête HTTP/1.0
  - Le serveur envoie la réponse
  - Le serveur ferme la connexion : on parle de **non-persistance**.
  - Si le client doit demander un autre document au même serveur alors, il doit ouvrir une nouvelle connexion.

HTTP 1.1 vs HTTP 1.0

- **Persistance :**
  - La version 1.1 du protocole HTTP -contrairement à 1.0- est en mode persistant: Le serveur garde la connexion ouverte (pendant un certain timeout) même après avoir envoyé la réponse à une requête en attente d'autres.
  - Avantages :
    - ✓ Réduire le temps de réponse aux requêtes puisqu'on économise le temps d'ouverture de connexion;
    - ✓ Consommation réduite du temps CPU des routeurs et des hôtes (clients, serveurs, proxies ou caches);
    - ✓ Congestion réseau réduite;
    - ✓ Permet le Pipelining.

HTTP 1.1 vs HTTP 1.0

- **Pipelining :**
  - Pour améliorer davantage le temps de réponse, les clients HTTP1.1 implémentent par défaut le pipelining :
    - ✓ Pour chaque objet référencé dans le document HTML reçu, une requête est envoyée sans attente de la réponse de la précédente.
    - ✓ Le serveur répond toujours dans l'ordre des requêtes.
  - Avantage :
    - ✓ Une connexion TCP peut ainsi être utilisée plus efficacement.

Cookies

- **Intérêt :**
  - HTTP étant un protocole sans état, les applications web utilisent et attribuent un identificateur unique à un client (navigateur) se connectant sur le site web pour pouvoir **retrouver des informations sur l'utilisateur ou la session** lors d'une prochaine connexion.
  - Cet identificateur est envoyé au client dans un **cookie**
  - Ce cookie sera renvoyé au serveur Web la prochaine fois que le client se reconnecte sur le site web, permettant au serveur d'identifier la session et de répondre en conséquence.

Cookies

- **Principe :**
  - Un cookie se base sur les entêtes HTTP 'SetCookie' et 'Cookie' :
    - ✓ Réponse du Serveur : SetCookie : User ID=FFA23C45BB1,…
      - Le serveur crée la variable User ID et l'initialise
      - Cette information est placée dans un cookie de l'utilisateur
    - ✓ Requête Client : Cookie : User ID=FFA23C45BB1…
      - A chaque requête suivante envoyée au même serveur, le client adresse également les différentes variables du cookie
  - Quelques limites (cela dépend des navigateurs):
    - ✓ La taille maximale d'une cookie est de 4Ko
    - ✓ Le navigateur ne peut stocker plus de 300 cookies
    - ✓ Un serveur unique ne peut créer plus de 20 cookies par utilisateur

Cookies

- **Options :**
  - Expires : Indique la date d'expiration du cookie
    - ✓ Valeur : Date
    - ✓ Format : Jour,JJMMAAAAHH:MM:SSGMT
  - Domain : Définit le champ d'utilisation du cookie
    - ✓ Valeur : nom de domaine contenant au moins deux points
    - ✓ Le domaine est toujours identique à celui du serveur (Exp: .monsite.com)
    - ✓ Valeur par défaut : nom du serveur
  - Path : Réduit le champ d'utilisation du cookie à un répertoire
    - ✓ Valeur: un répertoire
  - Secure : Indique si le cookie peut être utilisé sur des sessions non SSL

VI. Sécurité des applications Web

1. Introduction
2. HTTP en bref
3. Principales attaques sur les applications Web
4. Firewall applicatif Web
5. Authentification des utilisateurs

VI.3. Principales Attaques

- **Classification:**
  - La plupart des attaques applicatives peuvent être classées dans l'une de ces deux catégories :
    - ✓ Attaques visant à compromettre la confidentialité, l'intégrité ou la disponibilté des resources de l'application :
      - Exemple : Buffer Overflow, SQL injection
    - ✓ Attaques visant à compromettre la 'relation de confiance' entre l'application et ses utisateurs :
      - Exemple : Cross-site Scripting, Cross Site Request Forgery

VI.3. Principales Attaques

- **Attaques les plus fréquentes :**
  - Manipulation des champs d'un formulaire (En général, les champs cachés d'un formulaire).
  - Détournement de la validation des paramètres utilisateur, par :
    - ✓ SQL Injection (exp, exécution d'une requête SQL pour accéder à une base de données via le formulaire d'une application Web)
    - ✓ Command Injection (exp, **via l'URL**), …
  - Cross-site Scripting (XSS) : Attaquer la relation de confiance entre l'application Web et l'utilisateur victime en faisant exécuter à son navigateur un code malicieux (javascript, flash, Activex, …) suite à la connexion au site Web de l'application vulnérable.
  - Cross Site Request Forgery (XSRF) : injection de requête illégitime par rebond en provoquant l'envoi de requêtes par une victime vers un site vulnérable à son insu et avec son identité.
  - Vol d'identité ou de session (exp : Vol de Cookies via une attaque XSS), …

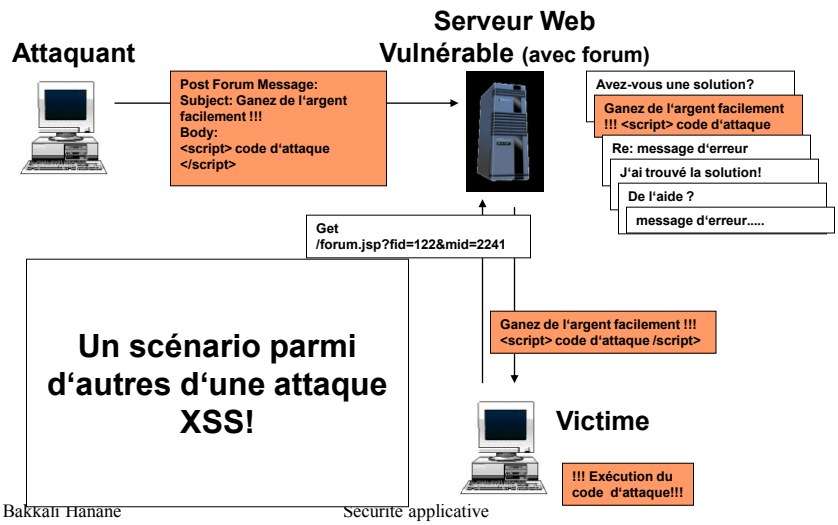
VI.3. Principales Attaques

- **Exemple d'une attaque SQL injection**

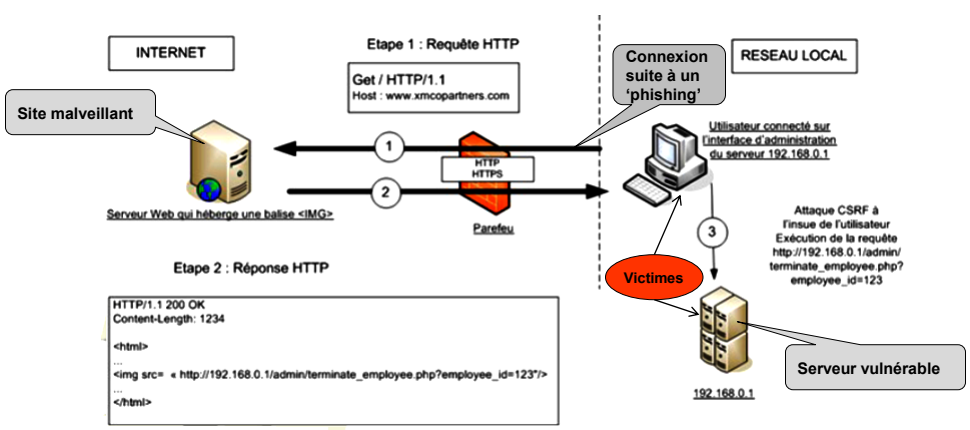
GET /user_lookup.cfm? lastname=foo';DELETE FROM users WHERE 1='1 HTTP/1.1

SELECT *
FROM users
WHERE lastname = 'foo';DELETE FROM users WHERE 1='1'

- Exemple d'une attaque XSS



- Exemple d'une attaque XSRF :



VI.3. Principales Attaques

- **Principales Mesures de protection (1/2):**
  - Gérer les autorisations sur la base de données (exp : contrôle d'accès basé sur les rôles)
  - Opter pour le principe du «Least priviliges» pour les accès aux bases de données par les utilisateurs (exécution très limitée de SELECT, UPDATE, DELETE).
  - S'assurer que l'application critique requiert une **authentification** qui n'est pas celle générée automatiquement par le navigateur (cookies).
  - Gérer les messages d'erreurs pour éviter la prise d'empreinte.
  - Validation des paramètres et des données utilisateur :
    - ✓ Tous les paramètres doivent répondre à des contraintes strictes (taille maximale, type, caractères valides, liste blanche plutôt qu'une liste noire);
    - ✓ Il est important durant la phase de design applicatif, de bien cibler quel paramètre pourra comporter des caractères spéciaux;
    - ✓ Ne pas compter sur la validation côté client.

VI.3. Principales Attaques

- **Principales Mesures de protection (2/2):**
  - Faire des tests de vulnérabilités aux applications et serveurs Web :
    - ✓ Utilisation des proxies (Man in the Middle) comme Paros ou Webscarab;
    - ✓ Utilisation des scanneurs comme Nikto
  - Renforcer la sécurité des OS (Hardening) .
  - Ne pas compter entièrement sur les Firewalls applicatifs (attention à la publicité des vendeurs).
  - Ne pas oublier les antivirus (et leur mise à jour).

VI. Sécurité des applications Web

1. Introduction
2. HTTP en bref
3. Principales attaques sur les applications Web
4. Firewall applicatif Web
5. Authentification des utilisateurs

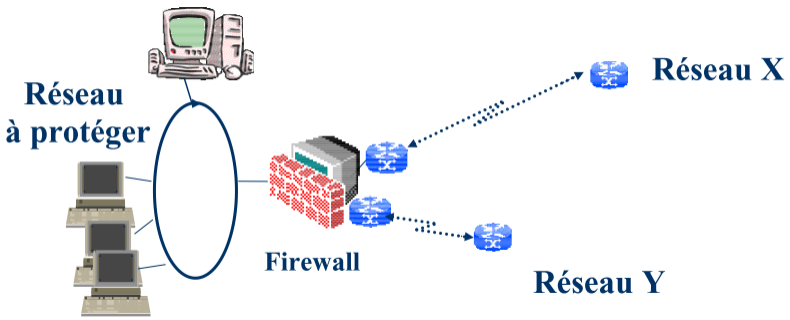
VI.4. Firewall Applicatif (WAF)

➢ **Description:**

- Le firewall applicatif Web ou WAF(Web Application Firewall) permet de :
  - ✓ bloquer le trafic applicatif suspect ou indésirable (entrant et sortant)
  - ✓ cacher les détails de l'infrastructure applicative du regard du monde extérieur.
- Un WAF vise donc à protéger ainsi les applications Web contre les attaques externes et les fuites de données internes.
- La mise en place d'un WAF doit refléter la **politique de sécurité** de l'entreprise.
- Si l'entreprise ne dispose pas d'une bonne politique de sécurité, le plus performant des WAF restera inefficace.
- En plus du renforcement de la sécurité des applications Web, un produit WAF offre généralement des outils de gestion de la disponibilité et des performances.

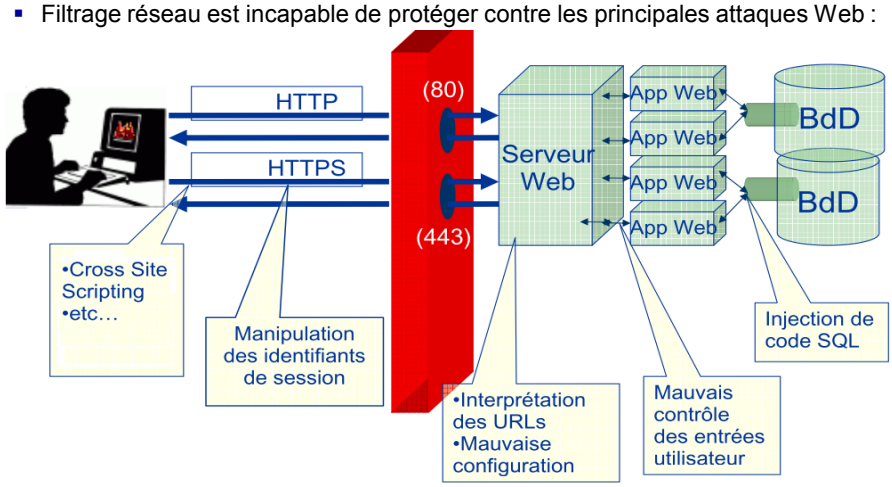
VI.4. Firewall Applicatif (WAF)

- **Firewall Réseau VS Firewall Applicatif :**
  - Filtrage basé essentiellement sur l'adresse IP et le n° de port.



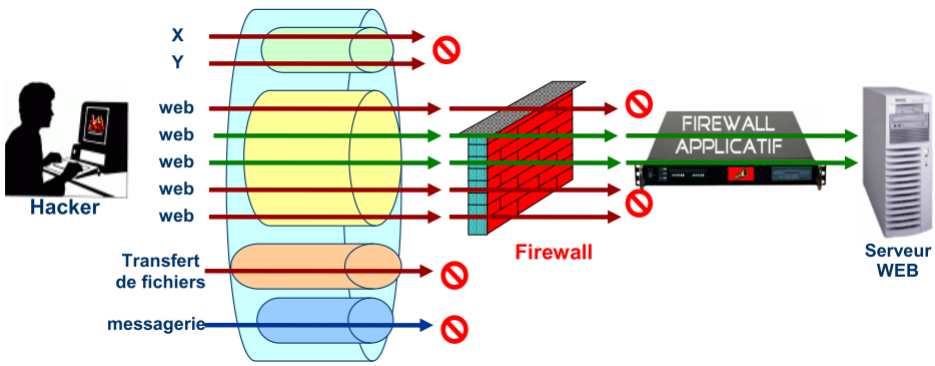
VI.4. Firewall Applicatif (WAF)

- **Firewall Réseau VS Firewall Applicatif :**
  - Filtrage réseau est incapable de protéger contre les principales attaques Web :



VI.4. Firewall Applicatif (WAF)

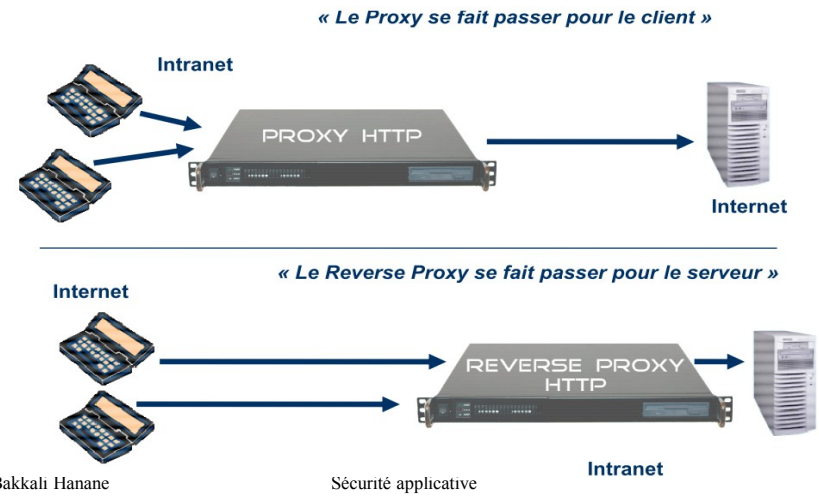
- **Pourquoi un Firewall Applicatif (Virtual patching) ?**



VI.4. Firewall Applicatif

- **Principales fonctionnalités :**
  - Compléter le filtrage réseau en apportant une protection contre les principales attaques Web (SQL Injection, Cookie poisoning, XSS, …)
  - Assurer le « Web Site Cloaking » (dissimulation des sites Web) en cachant les informations concernant l'architecture du site Web, les technologies et logiciels utilisés, les adresses des serveurs back-end…
  - Jouer le rôle d'un **Reverse Proxy** (dans le cas où il est configuré en mode Reverse Proxy)
  - Améliorer éventuellement les performances (Accélérateur SSL, Load balancing, …)
- **Exemples de WAF :**
  - Mod_security d'Apache (open source)
  - BeeWare (société française achetée en 2014 par DenyAll)
  - Leaders du marché américain: F5 A(pplication Security Manager), Imperva, Fortinet, Citrix, Acamai, …

- **Reverse proxy VS proxy :**



- **Critères de comparaison (1/2):**
  - Mode de déploiement : reverse proxy, transparent, SaaS ou autre
  - Appliance ou logiciel : Disponibilité d'accélérateurs SSL matériels
  - Haute disponibilité :
    - mode de diffusion de la configuration
    - synchronisation des différentes tables(connexions, authentifications
  - Support d'autres protocoles : LDAP, DNS, FTP, …
  - Authentifications supportées : Basic, Digest, Certificats, Méthodes à 2 facteurs…
  - Filtrage des réponses : «Status Code», entêtes, corps de réponse

VI.4. Firewall Applicatif

- **Critères de comparaison (2/2):**
  - Méthodes de détection
    - Liste noire et/ou liste blanche
    - Fourniture d'une base d'attaques régulièrement mise à jour
    - Détection des 'Zero-day attacks'
  - Protection contre les attaques
    - Par brute force,
    - Manipulation des cookies,
    - Modification des champs cachés, …
  - Enregistrements (logs) :
    - Informations enregistrées et modes d'enregistrement,
    - solution de reporting.

ENSIAS

VI.4. Firewall Applicatif

- **Limitations**
  - Lourdeur de mise en oeuvre (Time-consuming) notamment en ce qui concerne la création des signatures /règles de blocage des attaques (basées parfois sur des Dunamic AST).
  - Problème de scalability.
  - Choix du mode de déploiement (facilité VS efficacité)
  - Mais, parfois exigé (comme par PCI-DSS V3).

Introduction

- **Rôle d'un mécanisme d'authentification :**
  - Permettre à une entité de fournir la preuve de son identité à une autre entité
    assurer l'authentification d'une entité auprès d'une autre.
- **Types de preuves :**
  - Information connue 'seulement' de l'entité à authentifier et celle authentifiante, par exemple, un mot de passe ou une clé secrète partagée.
  - Information qu'elle possède, en général, un certificat et la clé privée correspondante.
  - Objet qu'elle est la seule à posséder, par exemple, une carte à puces.
  - Caractéristique physique propre, par exemple, empreinte biométrique.

Introduction

- **Niveaux d'authentification :**
  - Niveau 0 : Identification seule sans preuve
    - ✓ "Je dis qui je suis".
  - Niveau 1 : Identification et fourniture d'un mot de passe.
    - ✓ "Je dis qui je suis et ce que je sais".
  - Niveau 2 : Identification et fourniture d'un objet physique ou d'une information possédée (exemple, carte à puce ou certificat) plus éventuellement un mot de passe.
    - ✓ "Je dis qui je suis, ce que je sais et je donne ce que je possède".
  - Niveau 3 : Identification et fourniture d'une preuve physique (empreinte digitale, fond de l'œil, signature vocale) plus éventuellement un mot de passe.
    - ✓ "Je dis qui je suis, ce que je sais et je prouve physiquement qui je suis".

Méthodes d'authentification

- **Authentification par mot de passe simple (1/2):**
  - Le moyen le plus classique et le plus utilisé pour l'authentification.
  - Il permet d'offrir un premier niveau de sécurité
  - Il doit être connu non seulement par son 'propriétaire' mais aussi du vérificateur (exp. Un serveur), mais il doit rester secret pour les autres.
  - **Principe** :
    - ✓ Pour être authentifié auprès d'un vérificateur, le propriétaire du mot de passe commence d'abord par révéler son identité (exp. Nom d'utilisateur);
    - ✓ Il fournit ensuite son mot de passe au vérificateur qui le compare avec la copie dont il dispose.

Méthodes d'authentification

- **Authentification par mot de passe simple (2/2):**
  - Un bon mot de passe doit :
    - ✓ Comporter au minimum 14 caractères (de préférence plus) ;
    - ✓ Alterner caractères de contrôle, de ponctuation, chiffres, ainsi que les majuscules et minuscules ;
    - ✓ Être difficilement déduit ou deviné (exp. appartenance à un dictionnaire) ;
    - ✓ Être mémorisable (puis mémorisé en mémoire !) ;
    - ✓ Être protégé (exp. véhiculé et enregistré de manière cryptée et intègre) ;
    - ✓ Être réservé à un seul contexte (exp. un mot de passe/application) ;
    - ✓ Être fréquemment modifié (exp. chaque trimestre);
    - ✓ …

Méthodes d'authentification

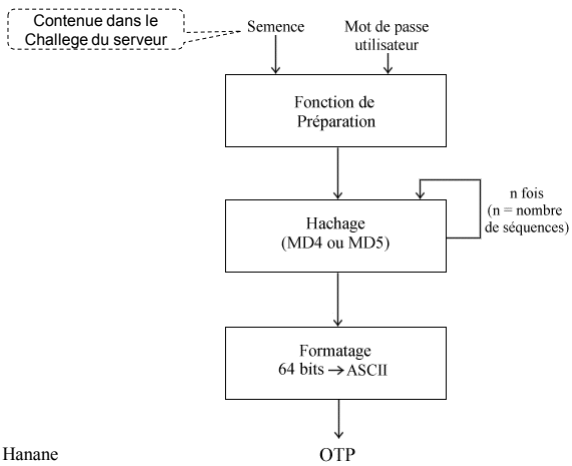
- **Authentification par mot de passe à usage unique (1/4):**
  - **Description :**
    - ✓ Un mot de passe à usage unique (one time password ou OTP) est un moyen d'authentification plus fort que le mot de passe classique.
    - ✓ Il est basé sur le principe de défi/réponse (challenge/response).
    - ✓ Il ne peut être utilisé que pour une et une seule session.
    - ✓ Il n'est plus choisi par l'utilisateur mais généré automatiquement, toutefois en se basant sur une valeur initiale choisie par l'utilisateur (ie. un autre mot de passe!!).

Méthodes d'authentification

- **Authentification par mot de passe à usage unique (2/4):**
  - **Génération :**
    - ✓ La génération d'un OTP se fait en trois étapes :
      - Une préparation, qui va prendre en compte le mot de passe utilisateur et la semence extraite du Challenge envoyé par le serveur,
      - La génération, qui consiste à appliquer une fonction de hachage n fois sur le résultat de la préparation, n étant le nombre de séquences.
      - Le formatage du résultat précédent de longueur égale à 64 bits en un mot de passe OTP à base de caractères ASCII.

Méthodes d'authentification

- **Authentification par mot de passe à usage unique (3/4):**



Méthodes d'authentification

- **Authentification par mot de passe à usage unique (4/4):**
  - **Avantages :**
    - ✓ Le mot de passe est utilisé une seule fois →pas de problème de longévité du mot de passe.
    - ✓ Le mot de passe est calculé automatiquement → pas de problème de mémorisation du mot de passe OTP (Mais le mot de passe utilisateur doit être enregistré de manière crypté).
    - ✓ Le carackage du mot de passe (par dictionnaire ou par force brute) n'a plus de sens et ne présente plus de risques.
    - ✓ Le mot de passe à usage unique peut être envoyé en clair sur le réseau car même s'il serait intercepté, il n'est plus réexpoitable.

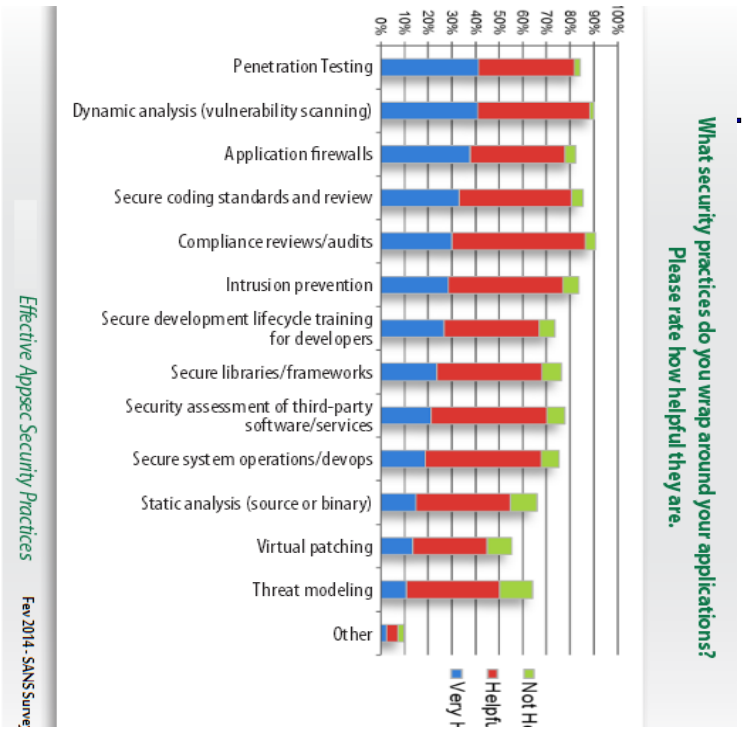
Méthodes d'authentification

- **Authentification SSO (Single Sign On) :**
  - **Principe:**
    - ✓ Pour chaque utilisateur : un seul mot de passe pour toutes les applications,
    - ✓ Un SSO gère :
      - Un ensemble d'utilisateurs (annuaire)
      - Un ensemble d'applications
  - **Fonctionnement :**
    - ✓ Inscription de l'utilisateur dans l'annuaire,
    - ✓ Vérification des droits d'accès lors d'une tentative d'accès à une application,
    - ✓ Décision en matière d'authentification.

Méthodes d'authentification

- **Authentification SSO (Single Sign On) :**
  - **Avantages :**
    - ✓ Plus convivial pour les utilisateurs
    - ✓ Faciliter l'évolution des méthodes d'authentification,
    - ✓ Gérer les droit d'autorisation d'accès à une application,
    - ✓ Contrôler les accès,
    - ✓ Mutualisation (plusieurs applications servies),
    - ✓ Faciliter l'audit et le suivi des utilisateurs,…
  - **Attention : Risques liés à la centralisation !**

Perception de l'efficacité des mesures de sécurité



VI.6. Conclusion

Remarques finales

- ↗ Risques au niveau de la sécurité applicative
- ↗ Prise de conscience de l'importance de la sécurité applicative
- Mesures de sécurité variés existent (outils, méthodes, …) mais dont l'efficacité reste à améliorer
- Sécurité réactive (après incident) plutôt que préventive ou à la source: besoin de méthodes de développement sécurisé ou security by design.
- Importance de la formation d'ingénieurs en sécurité applicative, et la dotation des développeurs et des managers de compétences en sécurité

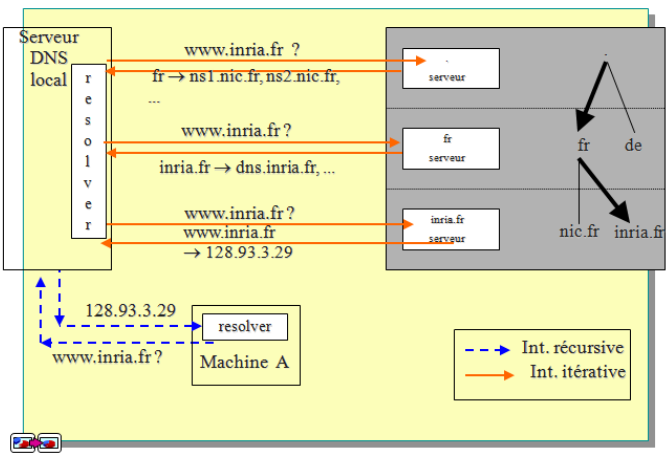
Sommaire du cours

VII. Sécurité des protocoles applicatifs

1. Sécurité du DNS
2. Sécurité de la messagerie

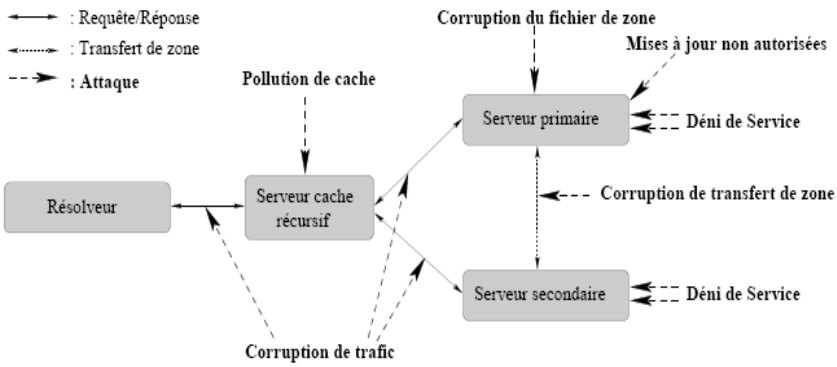
VII.1. Sécurité du DNS

Résolution de nom DNS



Quelques attaques possibles

Test d'un serveur DNS (http://dnsstuff.com/dnsreport/)



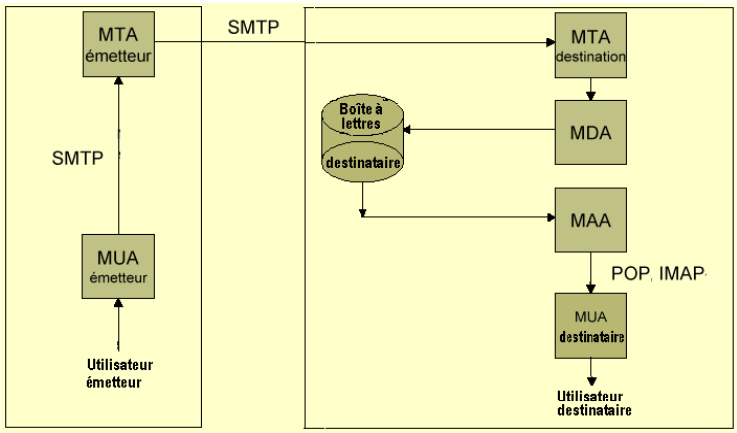
TSIG/TKey

- **LeTransfert de zone par TSIG/TKEY :**
  - ✓ Les données du fichier de zone sont d'abord hachées (par un algorithme de hachage) puis cryptées par un algorithme de cryptage symétrique en utilisant une clé partagée entre le serveur principal et le serveur secondaire.
  - ✓ Le résultat est une sorte de signature qui permet d'assurer l'intégrité et l'authentification du fichier de zone transféré (les données sont bien celles émises par le bon serveur autoritaire).

DNSSEC

- **Principe :**
  - ✓ Chaque zone sécurisée possède une paire de clés (clé publique et clé privée). La clé publique de la zone fille est signée par la clé privée de la zone parente (ou par une autorité de certification), sauf la clé de la racine qui est auto signée par elle-même.
  - ✓ La clé privée d'une zone sert à signer les RRs du fichier de zone.
  - ✓ DNSSEC définit de nouveaux enregistrements de ressources afin de contenir les clés et les signatures numériques : KEY RR, SIG RR, NXT RR et DS RR.
  - ✓ Lors d'une résolution de noms sécurisée, le résolveur doit vérifier la signature de la ressource demandée ainsi que les signatures des clés utilisées et des RRs DS qui les identifient. Le processus s'arrête lorsque le résolveur reçoit une clé qui est une clé de confiance.

Architecture



Principaux Risques

- Risques liés aux mails « légitimes » :
  - ✓ Perte de mail
  - ✓ Perte de confidentialité ou d'intégrité
  - ✓ Usurpation d'identité de l'expéditeur
  - ✓ Introduction d'un code malicieux/virus dans le corps du message ou en pièce jointe
  - ✓ **SPAM (un vrai fléau)**
  - ✓ Attaques DOS sur les serveurs de messagerie
  - ✓ Utilisation d'un serveur de messagerie comme open relay

Quelques mesures de protection

- Lutter contre les SPAM
  - ✓ Filtres anti-spam (Attention aux faux-positifs)
- Lutter contre les virus et les malwares
  - ✓ Antivirus et autres filtres de codes exécutables
- Assurer l'intégrité, la confidentialité ou l'authentification en utilisant :
  - ✓ S/MIME (Secure/ Multipurpose Internet Mail Extension)
  - ✓ PGP
  - ✓ Autres solutions
- Lutter contre la perte de messages :
  - ✓ Sauvegarde et archivage des mails importants (parfois pour des raisons légales)

➔ Idées à retenir :

- **Sécurité applicative** = un des maillons critiques de la **sécurité des SI**;
- Ce maillon est très lié aux autres de telle sorte qu'on ne le distingue pas toujours très bien des autres maillons (notamment de la sécurité réseaux et des données) ;
- **Le niveau de sécurité global d'une chaîne est celui du maillon le plus faible !**
- Il n'existe pas de sécurité **100%**, mais sans une bonne **politique de sécurité** il n'y a pas de bon projet de sécurisation.
  - ➢ Il faut d'abord connaître **les sources de risques** + les **ressources précieuses** avant de penser *Sécurité*.
- La sécurité doit avoir **un cycle vivant**, ne la laissons pas mourir → **Surveillance + Audit + mise à niveau.**
- Les **principes de base** de la sécurité **durent beaucoup plus** que les **outils de sécurité** → Veille technologique.
- **Sécurité ≅ *60%* (bonne politique de sécurité + sensibilisation) + *40%* (Outils: bien paramétrés + bien placés + bien exploités)**

IIX. Conclusion

➔ Liens Utiles (surtout pour la veille technologique):

- www.owasp.org
- www.net-security.org
- www.webappsec.org
- www.sans.org
- www.ssi.gouv.fr
- www.hsc.fr/ressources/index.html.fr
- www.clusif.asso.fr/fr/production/ouvrages/

IIX. Conclusion

➔ Quelques livres disponibles à la bibliothèque:

- Laurent Bloch et Christophe Walfhugel, **Sécurité Informatique : Principes et méthodes à l'usage des DSI,RSSI et administrateurs**, Eyrolles, 2009.
- Damien Seguy et Philippe Gamache , **Sécurité PHP 5 et MySQL**, Eyrolles, 2007.