Sécurité des Réseaux Informatiques

Micouille "The Boss" Riffon & Samuel "Big Boss" Monroe

30 Mai 2015

Table des matières

Chapitre 1

Avant-propos

1 The Boss

Si toi aussi ça te casse bien les couilles d'étudier dans des slides où les informations importantes concernant un sujet sont répartis sur 10-15 slides différents, voici le document qu'il te faut.

Sois redevable aux membres de F0XHOUND, toi qui galère à travailler sur une autre matière que TDS car on a presque quedalle à étudier cette année.

Ce document reprendra donc la matière non reprise dans le syllabus **mais** présente dans les slides (les 10-11-12). Et je précise que ce document reprend exactement la matière des slides avec parfois des précisions en plus dans la mesure du possible.

2 Big Boss

Ceci ne consitue pas une synthèse, mais plutôt une oeuvre composée de mes notes prises en cours, et de la réunion du syllabus de V.Van den Schrieck ainsi que du livre Computer Security de Stallings.

Le but est d'obtenir une compréhension totale du cours à la première lecture et d'offrir aux membres de Fox Hound les compétences qui sont attendues d'eux, afin de pouvoir pérenniser ce groupe et ses objectifs.

Chapitre 2

Généralités

Identification des différents concepts et principes de la sécurité informatique.

1 Concepts de sécurité informatique

Sécurité Informatique : Protection fournie à un SI automatisé pour atteindre les objectifs de préservation **CIA** des ressources (matériel, logiciels, données, etc..) du SI.

CIA est un concept clé dans la sécurité :

— **Confidentiality** : Préserve les autorisations et restrictions sur l'accès à l'information ainsi que sur la vie privée.

— **Integrity** : Protection contre la modification ou suppression impropre de données, incluant l'assertion de l'authenticité des données.

— **Availability** : Assure la disponibilité et accessibilité sûre aux données et services.

Ces trois concepts forment la triade **CIA**, aussi représentée en triangle, et indique les trois objectifs de préservation des assets (**data** & **services**).
Cette triade peut être additionnée de deux autres concepts :

— **Authenticity** : Propriété d'être authentique, et capable d'être vérifié et digne de confiance.

— **Accountability** : Besoin de pouvoir tracer les actions à des fins d'identification.

2 Terminologie

Voyons ici quelques termes reliés à la sécurité qu'il est nécéssaire de maîtriser :

2.1 Asset

Les assets, ressources, sont des éléments que les utilisateurs ou propriétaires souhaitent protéger.

Les assets peuvent être **hardware**, **software**, **données** et **réseux et moyens de communications**. Des ressources non physiques telles que la **réputation** doivent également être prises en compte.

2.2 Security Policy

Une **politique de sécurité** est un ensemble de règles ou pratiques qui spécifient ou régulent la manière dont un système ou organisation fournit des services de sécurité pour protéger les **assets**.

Composée de **contre-mesures**, ensemble de **règles** et de **procédures**.

2.3 Vulnerability

Faille ou faiblesse dans la conception, implémentation, opération ou gestion d'un système, qui peut être **exploitée** pour violer la **security policy**.

Ces vulnérabilités peuvent être la **corruption système** (I) , des **fuites** (C) ou des **indisponibilités** (A).

2.4 Threats and Attacks

Une **menace** est un danger potentiel de sécurité pour une ressource.

L'**attaque** est la réalisation de ce danger potentiel.
Elles peuvent êtres **actives** ou **passives** selon qu'elles altèrent le système ou non, et **internes** ou **externes** selon la position de l'attaquant par rapport à l'entreprise.

2.5 Countermeasures

La **contremesure** est une action visant à réduire une menace, vulnérability ou attaque via prévention, élimination.

2.6 Risk

Le **risque** est la prévision d'une perte exprimée par la probabilité qu'une menace particulière exploire une vulnérabilité particulière avec un résultat négatif particulier.

Un risque est **évalué** par la **probabilité que l'incident** survienne multiplié par **l'impact** que cet incident causerait.

3 Menaces, attaques et actifs

3.1 Divulgation non autorisée - Unauthorised Disclosure

Menace sur **Confidentiality**, attaques liées :

— **Exposition** : Exposition d'informations sensibles
— **Interception** : Interception d'informations dans une communication
— **Inférence** : Accession à des infos sensibles de manière indirecte
— **Intrusion**

3.2 Tromperie - Deception

Transmission de fausses infos à une entité en faisant croire que les infos sont conformes.

— **Masquerade** : Accès au système et agissement malicieux sous couvert d'une entité autorisée
— **Falsification** : Alteration de données valides
— **Répudiation**

3.3 Perturbations - Disruptions

Menaces sur **Availability** et **Integrity**.

— **Neutralisation** : Destruction physiques ou endommagement du matériel
— **Corruption**
— **Obstruction** : Interruption d'un service, DoS

3.4 Usurpation

Prise de contrôle d'un service ou système par une entité non autorisée.

— **Détournement** : Entité contrôle les ressoruces d'un système
— **Mauvais usage** : Composant système effectue une opération au détriment de la sécurité.

4 Principes de conception de systèmes sécurisés

Pas de techniques pour empêcher totalement des **failles**, mais des principes existent pour tendre vers un **système sécurisé** :

— **Economy of Mechanisms** : Solutions les moins complexes possibles pour éviter les failles (**KISS**).

— **Fail-Safe Default** : Comportement d'exclusion par défault
— **Complete Mediation** : Chaque accès système doit être validé, pas de caches
— **Open Design** : Elements de conception publiques
— **Separation of Privilege**
— **Least Privilege** : Droits minimum suffisant pour la tâche
— **Least Common Mechanisms** : Il faut minimiser les éléments partagés par les différents programmes ou utilisateurs.
— **Psychological Acceptability** : Les mécanismes de sécurité ne doivent pas interférer sans raison avec les tâches des users, qui pourraient les désactiver.
— **Isolation** : Systèmes publiques isolés des critques, processus et fichiers utilisateurs isolés de ceux des autres, accès aux mécanismes isolés
— **Encapsulation** : Isolation appliquée à l'orienté-objet
— **Modularity** : Partage des fonctions et services de sécurité pour une seule implémentation d'un prtocole de sécurité.
— **Layering** : Plusieurs couches de protection
— **Least Astonishment** : Une GUI doit se comporter comme l'attendrait un utilisateur

5 Stratégies de sécurité informatique

Trois étapes pour mettre au point une stratégie :

1. **Définir** la politique de sécurité
2. **Implémenter** la politique
3. **Valider** la politique

5.1 Politiques de sécurité

Il faut dans un premier temps définir les **assets** à protéger et leur **valeur** qui peut être définie en termes d'objectifs **CIA**.

Ensuite, il faut identifier les **vulnérabilités** et **attaques** potentielles, ainsi que leur **probabilité** de survenir, et l'**impact** qu'elles causeraient.

Pour chaque **vulnérabilité**, on peut mettre en place un série de contre-mesures, dont le choix s'établit sur des compromis tels que la **facilité d'utilisation et la sécurité**, ainsi que le rapport entre **coût de la mesure et coût des pertes**.

Les policies sont une décision **stratégique importante**, chaque choix est un renoncement.
Il faut lister les risques pour lesquels de conte-mesures ne sont pas prises, appelés **risques résiduels**.

5.2 Mise en place des contre-mesures

Implémentation des mécanismes de sécurité impliques quatre groupes d'actions complémentaires :

1. **Prévention** : Permet de restreindre le nombre d'attaques potentielles
2. **Détection** : Pas toujours possible d'éviter une attaque, mais il faut pouvoir la détecter pour prendre action
3. **Réaction** : Que faire en cas d'une attaque découverte pour en minimiser les conséquences
4. **Récupération** : Comment réagir suite à une perte (backups, etc...)

5.3 Validation

Le bon fonctionnement du système de sécurité doit être évalué et analysé, il faut donc établir des procédures de validation pour ce faire.

6 Connaître Son Ennemi

Important de connaître l'adversaire pour savoir ce qu'il recherche, ses buts et méthodes et s'en protéger.

6.1 Types d'attaquants

— **Attaques Automatisées** : Virus et vers conçus pour se propager automatiquement sur internet.
— **Scripts Kiddies** : Amateurs utilisants des scripts afin d'expérimenter.
— **Vandale** : Attaquant souhaitant montrer ce dont il est capable pour sa réputation.
— **Activiste** : Attaquant expérimenté ciblant des entités pour raisons politiques.
— **Expert** : Tente de mettre en évidence des failles en réalisant les intrusions, pas de but de destruction.
— **Espion** : Professionel disposant d'outils et méthodes sophistiqués, attaques sur de longues périodes.
— **Cybercrime Organisé** : Organisations à buts financiers, volent par exemple les numéros de cartes de crédit.
— **Cyberterrorisme** : Groupes d'experts bien organisés, buts terroristes et de destruction.

6.2 Methodes d'attaques

Attaquent des cibles **aléatoires** ou **spécifiques**.

Les cibles **spécifiques** nécéssitent une préparation minutieuse, celles **aléatoires** sont parfois utilisées comme point d'attaques par rebond.

6.3 Outils privilégiés

Les outils utilisés servent principalement à la collecte d'information.

— **WHOIS** : Infos sur Ip, Domain Name
— **Mailing Lists** : Consultation d'archives avec descriptifs de certains problème et infos techniques précises
— **Social Engineering** : Ingénierie Sociale, le but est d'exploiter les failles humaines afin d'en retirer des informations
— **Port Scan** : Montre les services ouverts
— **OS Fingerprinting** : Permettent d'identifier le type d'OS sur base de réponses, ainsi que versions des services etc, utile pour trouver une cible appropriée pour une faille sur une certaine version d'un logiciel.

7 Exercices

7.1 Questions de révision

— **Définition de la sécurité informatique :**

Protection fournie à un système d'information automatisé pour atteindre les objectifs de préservation de CIA (Confidentiality, Integrity, Availability) des ressources du système d'information (matériels, logiciels, firmwares, données, télécommunications).

— **Différence entre attaques passives et actives :**

Active : L'attaque essaie d'altérer le système et ses opérations.
Passive : L'attaquant essaie d'obtenir ou d'utiliser les informations du système sans affecter les ressources.

— **Listez et définissez brièvement des attaques actives et passives :**

— **Man in the Middle - Passive** : Attaque ayant pour but d'intercepter les communications entre deux parties sans que ni l'un ni l'autre ne puissent se douter que le canal de communication est compromis.

— **Keylogger - Passive** : Type de spyware spécialisé pour espionner les frappes clavier sur l'ordinateur hôte, et pour les transmettre via internet à un pirate pour qu'il les exploite.

— **Injections SQL - Actif(Passif)** : Type d'exploitation d'une faille de sécurité d'une application interagissant avec une base de données, en injectant une requête SQL non prévue par le système et pouvant compromettre sa sécurité.

— **Phishing - Passive** : Technique utilisée par des fraudeurs pour obtenir des renseignements personnels dans le but de perpétrer une usurpation d'identité, ou d'utiliser malicieusement ces informations personnelles (vols d'informations bancaires).

— **Denial Of Service - Actif** : Attaque informatique ayant pour but d'empêcher les utilisateurs légitimes d'un service de l'utiliser : Innondation d'un réseau, obstruction d'accès à un service à une personne donnée, envoi de milliards d'octets vers une cible.

— **Etapes de mise en place d'une stratégie de sécurité informatique :**

1. **Définir** la politique de sécurité, définir ce que le mécanisme doit faire.

2. **Implémenter** cette politique, quels mécanismes mettre en place.
Analyser le **Trade-off** (Coût/Bénéfice), prévention, détection, réaction et récupération, documenter les risques résiduels et assurer la maintenance.

3. **Valider** cette politique, s'assurer que le mécanisme fonctionne.

— **Selon quels critères va-t-on sélectionner les contre-mesures à appliquer dans le cadre d'une politique de sécurité informatique ?**

— Le premier choix à faire est un compromis entre la facilité d'utilisation et la sécurité.

— Deuxièment, il faut évaluer le coût que représenteraient une contre-mesure par rapport aux coût des pertes éventuelles et de la procédure de récupération.

8 Questions de réflexion

8.1 Exigences d'un distributeur de billets en termes CIA

1. **Availability :** Exigence basse, l'indisponibilité du service représenterait au pire un dérangement minimal pour l'utilisateur, car il peut toujours payer ses achats avec sa carte en magasin, manipuler son compte via e-banking, ou éventuellement aller retirer son argent dans un autre ATM d'une autre banque.

2. **Integrity :** Exigence haute, car une erreur de données au niveau du retrait ou du dépôt de billets pourrait être catastrophique pour l'utilisateur, par exemple si son compte était débité de 500 euros pour un retrait de 50.
   L'inverse est valable pour la banque elle-même, dans un cas où les utilisateurs recevraient beaucoup plus que le montant demandé.

3. **Confidentiality :** Exigence haute, une faille dans la confidentialité serait catastrophique si quelqu'un de mal intentionné pouvait avoir accès au compte de l'utilisateur.
   Cette faille aurait aussi un impact considérable sur la réputation de la banque.

8.2 Impact d'attaques en termes de CIA sur :

— **Organisation avec mise à disposition d'info publiques sur son serveur Web :**

1. C : Documents publiques, pas de réel impact puisque disponibles pour tout le monde.
2. I : Impact faible, les documents étant publiques, un impact sur l'intégrité pourrait gêner les utilisateurs et les possesseurs.

3. A : Impact faible à modéré, tout dépend de l'utilité de ces documents au public, mais si ces données sont publiques c'est qu'elles sont probablement de moindre importance et leur indisponibilité serait au pire gênante pour l'utilisateur.

— **Organisation policière, données exrêmement sensibles pour investigations :**

1. C : Les documents étant extrêmement sensibles, l'impact est élevé, toute divulgation pourrait nuire grandement au bon déroulement des enquêtes avec toutes les conséquences que cela entraînerait.

2. I : Leur intégrité est primordiale, pour les mêmes raisons que sus-mentionnées.

3. A : Leur indisponibilité pourrait avoir au pire un impact modéré, tout dépend des échéances de l'enquête.

— **Organisation financière, information administrative de routine :**

1. C : Impact modéré, ces informations étant de routines, on peut envisager que leur divulgation serait au pire assez gênante pour l'organisation, mais pas au point de pouvoir mettre en péril ses activités.

2. I : Impact élevé, même si ces informations sont de routines, des erreurs dans ces informations pourraient mener à des chiffres non justes dans les calculs finaux de l'organisation, l'intégrité doit être assurée absolument.

3. A : Impact modéré-élevé : Si ces informations sont dites "de routine", on peut estimer que leur indisponibilité empêcherait le fonctionnement en temps réel de l'organisation sur certains secteurs, situation qui pourrait être dommageable pour celle-ci.

— **Système SCADA - Organisation militaire**

1. C : Impact faible pour les informations de routine, tandis qu'une faille dans la confidentialité des informations de senseurs pourraient avoir un impact élevé si exploités par une nation ennemie ou un groupe terroriste.

2. I : Impact faible pour les informations de routine, probablement élevés pour celles de senseurs.

3. A : Impact faible pour informations de routine, élevés pour celles de senseurs.

8.3 Activités considérées comme menace potentielle pour le réseau d'une entreprise, et pourquoi ? :

— **Employé responsable de la distribution interne du courrier :** Menace potentielle, cet employé pourrait par exemple détourner le courrier à destination des supérieurs, ou à destination de certains secteurs sensibles de l'entreprise.

— **Anciens employés licenciés pour cause de restructuration :** Certains pouvant sans doutes êtres rancuniers vis-à-vis de la situation vécue, ils pourraient éventuellement essayer de se servir de leurs anciens logins pour causer du tort à l'entreprise, récupèrer des documents sensibles, etc...

— **Employé en voyage d'affaire :** Divulgation d'informations sensibles au cours d'une soirée arrosée.

— **Compagnie de gestion des bâtiments :** Les système d'extinction automatiques étant contrôlés par cette compagnie, et pas par l'entreprise, on ne connaît pas le niveau de sécurité de leur système informatique et de leur système de contrôle à distance, quelqu'un ayant accès à cette compagnie pourrait activer ces extincteurs, voir les désactiver.

Chapitre 3

Bases de Cryptographie

Outil de sécurité, étude des principaux algorithmes et champs d'applications ainsi que leurs propriétés.

1 Chiffrement Symétrique

1.1 Principes

Technique universelle pour assurer **confidentialité** des échanges ou stockage de données.

Il est composé de cinq éléments :

1. Message
2. Algorithme de chiffrement
3. Clé secrète
4. Message chiffré
5. Algorithme de déchiffrement

Deux conditions pour que le chiffrement symétrique soit **sûr** :

1. Agorithme **fort**, pas possible de reconstituer la clé.
2. Emetteur et receveur doivent garder la clé secrète en **sécurité**.

Le chiffrement symétrique est **vulnérable** à deux types d'attaques :

1. **Attaques cryptanalytiques** : Se base sur la nature de l'algorithme ou analyse de caractéristiques du message pour déduire la clé ou le message chiffré.

2. **Attaques brute-force** : Essais de toutes les combinaisons possibles.

1.2 Chiffrement Symétrique par Bloc

Consiste à découper le message en blocs chiffrés les uns après les autres, le message chiffré est la **concaténation** des blocs.
Algorithmes **DES** et **AES** fonctionnent de la sorte.

Data Encryption Standard

Algorithme **DEA** à appliquer à des blocs de **64bits** et à une clé de **56bits**, blocs chiffrés produits de **64bits**.

Vulnérabilités à deux niveaux :

— Attaques cryptanalytiques sur l'algorithme
— Clé de 56 bits vunérable aux attaques brute-force

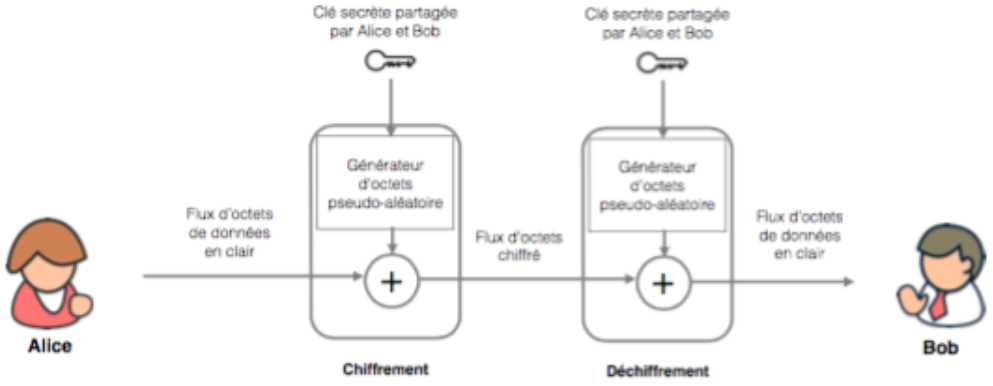
Vulnérabilités comblées en chiffrant plusieurs fois le message (**Triple DES**) créant une clé de 112 ou 168 bits, mais génère un **temps calcul** plus long.

Advanced Encryption Standard

AES, conçu pour être au moins aussi secure que **3DES** avec des blocs de **128bits** et supporte des clés de **128,192,256 bits**.

1.3 Chiffrement symétrique par flux

Travaille de manière continue en chiffrant **un octet** à la fois. A partir de la clé secrète et du flux de données en clair, génère un flux pseudo-aléatoire.



2 Fonctions de Hashage et authentication des messages

Chiffrement permet de garantir la **confidentialité** des échanges, mais d'autres mécanismes permettent de garantir **intégrité** et **authentifier** un message.

L'**authentification** des données nécéssite deux vérifications :

1. La source doit être **authentique**
2. Le message n'a pas été **altéré**

2.1 Authentification de message sans chiffement

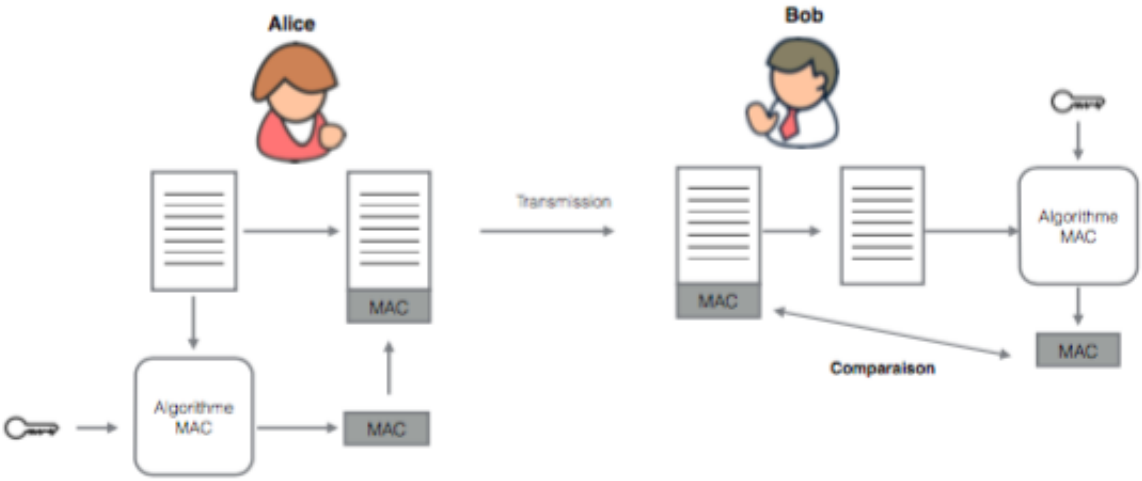
Idée : Générer un tag d'authentification et l'ajouter à la fin du message, parfois mieux de laisser le message en clair si le receveur est lourdement chargé.
L'authenticité est validable via un échantillon des données aléatoire.

Les programmes informatiques sont souvent **authentifié** de cette manière via un tag.

Code d'authentification de message

Principe du **MAC** est de générer un code sur base du **message** et d'une **clé secrète** partagée :

1. Le code est généré avec la clé secrète
2. Il est ajouté au message et transmis
3. Le receveur recalcule le **tag** et le compare avec celui reçu
4. Il donc constate si le message n'a pas été altéré



Avec ce mécanisme, on a les garanties suivantes :

— Receveur sûr que le message n'a pas été **altéré**
— Receveur sûr que le message vient d'un émetteur authentique puisque lui seul peut calculer le code
— Le message est reçu conformément à la séquence

DES utilisable comme algorithme pour **MAC**.

Fonctions de hashage unidirectionnelles

Alternative à **MAC**, le tag est généré sans clé secrète est est appelé **digets**.
Il assure l'intégrité du message uniquement, l'authenticité de la source doit être assurée par un autre moyen.

On parle de **signature digitale**.

Garanties de sécurité des fonctions de hashage

But : Fournir une empreinte digitale d'un bloc de données.
La fonction de hashage **H** doit posséder les propriétés suivantes pour être utilisée :

— Appliquable à un bloc de données de taille variable
— Résultat de longueur fixe
— H relativement léger à calculer, facilement implémentable
— Pas possible de retrouver le bloc de données sur base du tag
— Pas deux tags identiques pour deux messages différents

Les fonctions de hashage sont potentiellement **vulnérables** à la cryptanalyse et attaques brute-force, la longueur du hash est déterminante sur ces points.

MD5 produit des hash 128 bits dépréciés, SHA-3 est préconisé et utilisé pour le stockage sécurisé de mots de passes.

3 Chiffrement Clé Publique

Innovation avec **Diffie-Hellman**, chiffrement sur base d'opérations mathématiques.
Cette cryptographie est **asymétrique**, nécéssitant **2** clés.

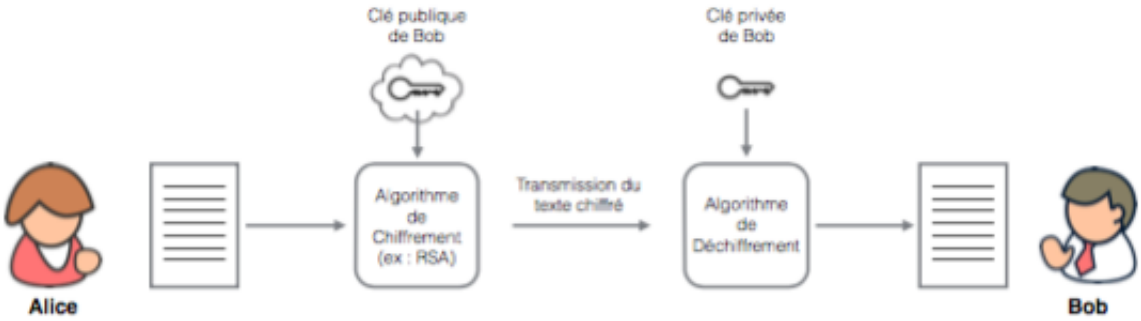
Pas plus sécurisé que le chiffrement symétrique, sa sécurité dépend de la longueur de la clé, et de la complexité de l'algorithme.
La distribution des clés est également un point sensible, et le chiffrement symétrique complémentera ce soucis du chiffrement asymétrique.

Le chiffrement asymétrique se base sur six éléments :

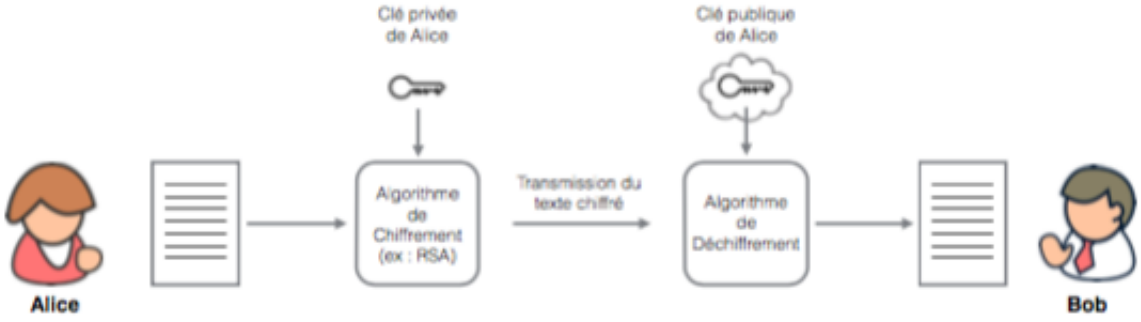
1. Message en clair
2. Algorithme de chiffrement
3. Clé publique, diffusée publiquement et utilsiée piur le **chiffrement**
4. Clé secrète, gardée secrète par son propriétaire et utilisée pour le **déchiffrement**
5. Message chiffré
6. Algorithme de déchiffrement utilisant la clé secrète et le message chiffré

Les étapes principales du chiffrement sont les suivantes, avec un schéma explicatif :

1. Les users créent une paire de clés privée/publique
2. Ils placent leurs clés publiques dans un registre ou fichier ; la clé secrète est gardée privée
3. Alice chiffre son message pour Bob avec la clé privée de Bob
4. Bob déchiffre le message avec sa clé privée



Un autre cas d'utilisation réside dans **l'authentification**, Alice chiffre le message avec sa clé privée, et le message originel est déchiffrable avec sa clé publique :



En pratique, le système **asymétrique** peut être utilisé dans trois catégories d'applications :

1. Signature numérique
2. Distribution de clés symétriques
3. Chiffrement de clés secrètes

Les conditions suivantes doivent être remplies par un algorithme de chiffrement symétrique :

— Possibilité de générer la paire en un temps de calcul raisonnable
— Chiffrement en temps raisonnable
— Déchiffrement en temps raisonnable
— Impossible en temps de calcul de déterminer la clé privée
— Impossible de recomposer message via clé publique
— Possibilité d'utiliser la publique ou privée pour le chiffrement, l'autre pour le déchiffrement

3.1 Algorithmes de chiffrement asymétriques

RSA

Plus couramment utilisé pour chiffrement à clé publique, chiffrement par **bloc**.
Permet la signature numérique, la distribution de clé symétrique et le chiffrement de clé secrète.
Longueur recommandée de **1024bits**.

Diffie-Hellman

But : Permettre échange de secret partagé entre utilisateurs.
Permet la distribution de clé symétrique.

DSS

Fournit un système de signature digitale via SHA, utilisable uniquement pour la signature numérique.

ECC

Cryptographie par courbe elliptique.

4 Signatures Digitales

On peut utiliser le chiffrement à des fins d'**authentification**, ceci peut être fait de manière légère en calculant une valeur de hashage sur le message, par exemple un hash **SHA-512**.

Cette opération consiste en les étapes suivantes :

— **A** calcule le hash de son message en SHA-512
— **A** chiffre le hash avec sa clé privée, cela crée la **signature digitale**
— **A** envoie le message+signature
— **B** déchiffre le hash avec la clé publique de **A**
— **B** recalcule le hash du message reçu en SHA-512
— **B** compare les deux hash pour avoir la certitude que **A** a bien envoyé le message

5 Gestion des clés

5.1 Certificats de clés publiques

Comment s'assurer que la clé publique reçue est bien celle de la personne qui s'annonce et pas un usurpateur ?

Cette question est réglée via le système de **certificats**
L'utilisateur associe sa clé publique à un identifiant, et va confier ce certificat à des authorités certificatives **CA** sont chargées de chiffrer celui-ci.

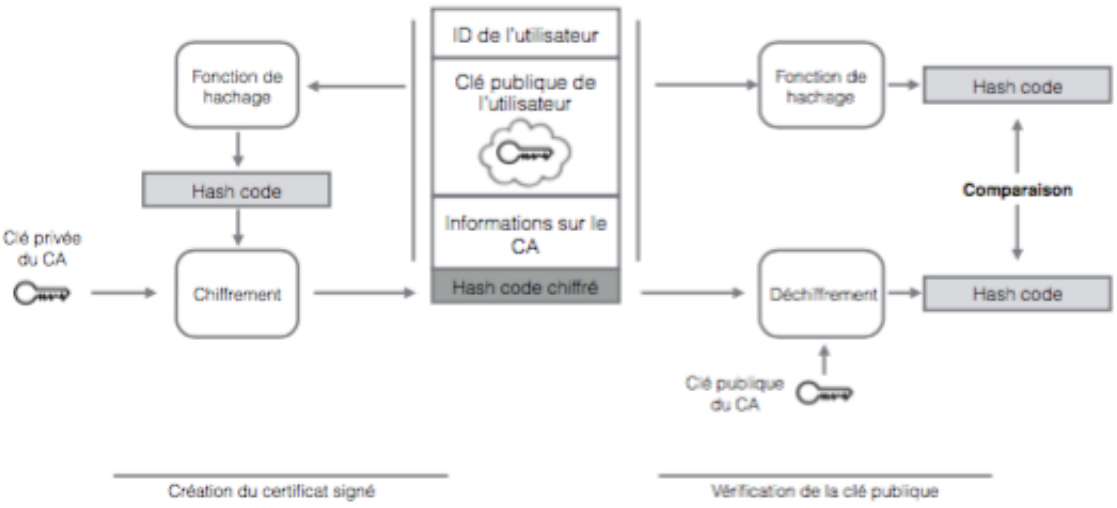
Le processus comporte plusieurs étapes :

1. Le user crée une paire de clés
2. Le user prépare un certificat non signé incluant son **identifiant** et sa **clé publique**
3. Le user fournit le certificat non signé à un **CA** de manière sécurisée
4. L'autorité certificative **CA** crée la signature numérique :

   — Calcule le hash du certificat
   — Chiffre le hash avec sa clé privée

5. Le **CA** attache la signature numérique au certificat non signé (**signature**)
6. Le **CA** renvoie le certificat signé au user
7. Le user peut maintenant fournir le certificat à qui le demande

Standard **X.509**



5.2 Echange de clés symétriques

La clé secrète étant **partagée**, il faut la sécuriser.
Cet échange sécurisé peut être fait via l'algorithme à clé publique (algo d'échange Diffie-Hellman).

5.3 Enveloppes numériques

Utilisation du chiffrement à clé **publique** pour protéger une clé **symétrique**.

— L'émetteur génère une clé secrète partagée aléatoirement
— Chiffrement du message avec la clé aléatoire
— La clé est-elle même chiffrée via clé publique du receveur et envoyée **avec** le message
— Le receveur déchiffre la clé avec sa clé privée, et peut déchiffrer le message avec la clé déchiffrée (spaghetti mofo).

6 Exercices

6.1 Questions de révision

— **Quels sont les éléments intervenants dans un chiffrement symétrique ?**

1. Le message en clair
2. L'algorithme de chiffrement
3. Une clé secrète appliquée à l'algorithme
4. Le message chiffré
5. L'algorithme de déchiffrement

— **Combien de clés faut-il pour que deux personnes puissent communiquer par chiffrement symétrique ?**

Il suffit d'une clé, qui est partagée entre les deux entités communiquantes.

— **Quelles sont les deux contraintes principales pour garantir la sécurité du chiffrement symétrique ?**

1. Un algorithme de chiffrement fort
2. L'émetteur et le reçeveur doivent avoir obtenu leur copie de clé de manière sécurisée

— **Listez les trois approches permettant de faire de l'authentification de messages :**

1. Le chiffrement des messages via le chiffrement symétrique
2. Ajout d'un tag en fin de message (MAC) calculé via clé symétrique
3. Ajout d'un tag via hashage unidirectionnel (signature digitale), plus besoin de clés partagées

— **Quels sont les éléments intervenant dans un chiffrement asymétrique ?**

— Le message en clair
— L'algorithme de chiffrement
— Une clé publique et une clé privée, une pour chiffrer et l'autre déchiffrer
— Le message chiffré
— L'algorithme de déchiffrement

— **Listez et expliquez brièvement trois utilisations d'un cryptosystème à clé publique**

1. La signature numérique : On génère le tag qui prouve la validité du message et on le chiffre avec la clé secrète, le receveur regénère ce tag avec sa propre fonction et vérifié le tag reçu en le déchiffrant avec la clé publique.
2. La distribution de clés symétriques : Si deux entités veulent se communiquer une clé secrète, Alice chiffre la clé secrète à partager avec la clé publique de Bob, Bob reçoit cette clé secrète et la déchiffre avec sa clé privée.
3. Le chiffrement de clé secrète : Afin de ne pas avoir sa clé secète en clair dans ses fichiers, Bob peut la chiffrer avec une autre clé publique.

— **Quelle est la différence entre une clé privée et une clé secrète ?**

La clé privée fait partie du système de chiffrement asymétrique, elle est utilisable pour déchiffrer des messages chiffrés avec la clé publique qui lui est paire, tandis que la clé secrète est conçue pour chiffrer **et** déchiffrer un même message.

— **Qu'est-ce qu'une signature numérique ?**

C'est un tag ajouté au message afin de prouvé sa validité, il est généré en calculant une valeur **hashée** du message, en chiffrant ce hash avec sa clé privée.
Ce hash chiffré accompagnera le message, sera déchiffrable en utilisant la clé publique de l'émetteur.
Le hash déchiffré sera comparé à un hash généré côté émetteur, leur correspondance prouvera que le message n'a pas été altéré.

— **Qu'est-ce qu'un certificat de clé publique ?**

C'est un certificat géré par une Autorité Certificative qui est composé de l'identifiant d'un tiers et de sa clé publique, et prouve son identité. Cela évite l'utilisation par masquarade d'une clé publique de quelqu'un d'autre.

— **Comment le chiffrement à clé publique peut-il être utilisé pour distribuer une clé secrète ?**

Alice chiffre la clé secrète avec la clé publique de Bob, Bob reçoit le message chiffré, lui seul peut le déchiffrer avec sa clé privée.
Bob et Alice possèdent maintenant tous deux la clé privée.

— **Faites un schéma expliquant le mécanisme d'enveloppe numérique**

TODO

6.2 Questions de réflexion

1. **On cherche un moyen de s'assurer que vous et une autre personne êtes en possession de la même clé secrète, sans transmettre cette clé. On vous propose de créer une chaîne de bits aléatoire ayant la longueur de la clé, d'effectuer un XOR entre cette chaîne et la clé secrète, puis d'envoyer le résultat à l'autre personne. Cette dernière effectue un XOR entre ce qu'il a reçu et sa clé secrète et renvoie le résultat. Vous pouvez alors vérifier que la chaine renvoyée correspond bien à la chaîne aléatoire que vous avez générée pour vous assurer que votre correspondant possède bien la même clé que vous. Que pensez-vous de la sécurité de ce mécanisme ?**

   Ce n'est pas sécurisé, en faisant un XOR des deux messages qui transitent entre les deux entités, on retrouve la clé secrète.

2. **Comment pourrait-on ajouter un mécanisme d'authentification au système d'enveloppe numérique ? Faites un schéma.**

   Génération de deux signatures digitales. **TODO**

Génération d'une signature digitale sur l'enveloppe elle-même.

Chapitre 4

Authentification de l'utilisateur

Première ligne de défense contre les attaques, est à la base des systèmes de contrôle d'accès et d'**accounting**.

Authentification : Processus de vérification de l'identité déclarée par un système, via **identification** et **vérification**.

1 Principes

L'authentification repose donc sur deux systèmes.

1. Premièrement un système où le user s'enregistre, se **crée un identifiant**.
   Ceci nécéssite la verification de l'identité, la création d'une structure de données liant l'identité à un token, et la remise du token au user.
   Le token peut être une clé de chiffrement ou un mot de passe.

2. Deuxièmement, un second système où le user veut s'authentifier via son identité et le toker reçu.
   Le token peut être :

   — Chose **connue** du user (mot de passe, pin)
   — Chose **possédée** par le user (carte accès, clé physique)
   — Chose **caractérisant** le user (empreinte digitale, rétinienne)
   — Chose **faite** par le user (graphologie, rythme au clavier, écriture)

2 Authentification par mot de passe

Technique de défense la plus utilisée, assure la vérification du **droit d'accès** du user et de ses privilèges.

Les **OS** tiennent une liste indexée des mots de passe par identifiant, les mots de passes sont stockés en hash.

Ces mots de passes sont la cible de nombreuses attaques, suivies d'une contre-mesure :

— **Attaque par dictionnaire hors-ligne** : Via accès au fichier des mots de passe, comparaison avec les valeurs de hashage d'un dictionnaire. Nécéssité de bloquer l'accès au fichier.
— **Attaque sur compte spécifique** : Tentative de se connecter sur un compte par essai de combinaisons. Bloquage de trop de tentatives échouées.
— **Attaque par mot de passe populaire** : Sensibilisation des utilisateurs.
— **Piratage de poste de travail** : Délog automatique des users après temps d'inactivité.
— **Social Engineering** : Exploitation d'erreurs humaines. Sensibilisation des utilisateurs.

— **Exploitation de l'usage multiple des mots de passe** : Sensibilisation des utilisateurs.
— **Surveillance électronique**

2.1 Utilisation des mots de passe hashés

Sous **UNIX** les mots de passes sont **hashé** avec une valeur de hashage de longueur fixe appelée **sel**.

Ceci permet d'obtenir des hash **différents** pour deux mêmes mots de passe en mémoire.

Lors de la connexion, le système récupère le hash associé au user, recalcule le hash **avec** ce sel et le mot de passe introduit, et compare les deux valeur de hash.

Ceci permet :

— Eviter mots de passe dupliqués visibles dans le fichier
— Complexifie les attaques par dictionnaire
— Impossible de savoir si une personne utilise les mêmes mots de passes sur ses comptes

2.2 Craquage des mots de passe

Approches traditionnelles

Développement d'un dictionnaire de mots de passe possibles et comparaison avec le fichier de mots de passe.
Chaque essai doit être combiné avec les valeurs de sels possibles.

Attaques **optimisables** via un fichier **rainbow-table** qui contient déjà les valeurs précalculées de hashs et leurs combinaisons, accélère sensiblement le crack.

Approches modernes

L'évolution des puissances de calculs permet à de simples machines d'effectuer beaucoup de travail. Des algorithmes de génération de mots de passes potentiels ont évolué et se basent sur de larges bases de données de mots de passes.

2.3 Contrôle d'accès au fichier de mots de passe

Nécessité de limiter l'accès au fichier de mot de passe, stockés dans un fichier séparés des utilisateurs appelé **shadow password**.

2.4 Stratégies de sélection d'un mot de passe

Compromis entre trop simple ou trop complexe pour les retenir, 4 bonnes techniques pour utilisation de bons mots de passes :

1. Sensibilisation des utilisateurs, guidelines pour mots de passes
2. Génération automatique de mots de passes
3. Vérification de mots de passes périodique via cracker
4. Vérification pro-active des mots de passes (lors de l'enregistrement)

3 Authentification par token

Objets physiques utilisés pour identifier un utilisateur.

3.1 Cartes à Mémoire

Stockage d'information, mais pas de traitement des données.

Contrôle d'accès physique ou associées à un code PIN.

Inconvéniants :

— Lecteur spécial nécéssaire
— Perte du token empêche l'utilisateur d'accéder au système
— Mauvaise expérience-utilisateur

3.2 Smart Cards

Nombreux types, catégorisés en quatre dimensions :

1. **Physique** : Carte bancaire, clé, calculatrice
2. **Interface Utilisateur** : Ecran et clavier pour interaction avec le user
3. **Interface Electronique** : Lecteur à contact ou sans contact
4. **Protocole d'authentification**

4 Authentification biométrique

Système se basant sur les caractéristiques physiques des individus, plus complexe et plus cher. Ce système n'est pas entièrement précis, et peut donner de faux-négatifs.

4.1 Caractéristiques physiques utilisées en biométrie

— Reconnaissance faciale
— Empreintes digitales
— Géométrie de la main
— Empreinte rétinienne
— Iris
— Signature papier
— Voix

5 Authentification à distance

Vulnérabilités supplémentaires car l'utilisateur doit utiliser un moyen de communication pour s'authentifier sur une machine distance, protocoles sécurisés **challenge/response**.

6 Attaques ciblant l'authentification

Les systèmes d'authentification sont vulnérables aux attaques suivantes :

— **Attaque Client** : L'attaquant se fait passer pour utilisateur légitime
— **Attaque Hôte** : Machine cible directement attaquée
— **Ecoute** : Keylogging, observation du user
— **Rejeu** : Ré-utilisation d'un échange d'authentification
— **Trojan Horse** : Application se faisant passer pour une app d'authentification authentique afin de captuer des logins
— **DoS**

7 Exercices

7.1 Questions de révision

— **Décrivez en termes généraux quatres moyens pour authentifier l'identité d'un utilisateur**

1. L'identification par mots de passes
2. Token
3. Biométrique
4. A distance

— **Listez et décrivez brièvement les menaces principales au secret des mots de passe**

— Attaque par dictionnaire hors-ligne : Accès au fichier mot de passe, craquage brute-force sur ce fichier (John The Ripper).
— Attaque sur un compte spécifique : Essais de combinaisons de logins/mot de passe.
— Attaque par mot de passe populaire : Essai de logins avec les mots de passes fréquemments utilisés.
— Piratage du poste de travail
— Social Engeneering
— Exploitation de l'usage multiple d'un mot de passe
— Surveillance électronique

— **Comment peut-on protéger un fichier de mot de passe ? Pourquoi est-ce important ?**

Premièrement via le hashage - salage des mots de passes, et ensuite en les stockant dans un fichier **shadow** situé dans les fichiers root, qui nécéssitent donc des droits d'administrations pour être accédés.
C'est important car un fichier de mots de passes peut-être utilisé avec un cracker dictionnaire pour en déloger les mots de passes hashés.

— **Listez et décrivez brièvement quatre techniques qui permettent de garantir que les mots de passes d'un système sont sécurisés**

1. Sensibilisation des utilisateurs
2. Génération automatiques des mots de passes
3. Vérification des mots de passes
4. Vérification proactive des mots de passes

— **Quelle est la différence entre une carte à mémoire et une smart card ?**

Carte-mémoire :
— Nécéssitent lecteur spécial
— Associé à un code
— Perte du token signifie risque de sécurité

Smart-Card :
— La plupart du temps génèrent un identifiant pour accèder au système, le token n'est donc pas l'objet direct d'authentification
— La perte de la smart-card n'implique pas de risques de sécurité

— **Listez les principales caractéristiques physiques utilisées pour l'authentification biométrique**

1. Reconnaissance faciale

2. Empreintes digitales
3. Géométrie de la main
4. Empreinte rétinienne
5. Iris
6. Signature manuscrite
7. La voix

7.2 Questions de réflexion

1. **Discutez de l'adéquation des mots de passes suivants :**

— 1-EBL-345 : Chiffres, lettres et caractères spéciaux, bon niveau
— mfmitm : Peu et niquement lettres bien que pas dans un dictionnaire, faible niveau
— Nathalie1 : Prénom + chiffre, très faible niveau
— Whashington : Variation d'un nom connu, faible niveau
— Aristotle : Nom connu tel quel, très très faible niveau
— tv9stove : Niveau moyen, pas de sens et présence de chiffres

Chapitre 5

Contrôle d'accès

1 Généralités

Garantir la protection des **assets** en contrôlant les accès à ceux-ci.

1.1 AAA

Contrôle d'accès à un système modélisé par le protocole **Authentication - Authorisation - Accounting**.

— **Authentication** : Vérification de la validité des éléments prouvant l'identité d'une personne ou d'une entité système.

— **Authorization** : Vérification des autorisations de l'entité identifiée par rapport aux autorisations requises.

— **Accounting** : Surveillance des accès aux ressources pour réaction en cas d'usurpations, via processus d'audit ou de logging.

2 Modèles de contrôle d'accès

Politiques de contrôle d'accès catégorisables en plusieurs politiques, pas mutuellement exclusives :

— **Discretionnary Access Control** : Règles fixant quels demandeurs sont ou non autorisés à effectuer une action donnée, appelé discretionnaire car une entité peut transmettre ses droits.

— **Mandatory Access Control** : Contrôle sur base du niveau de sécurité de la ressource, mandataire car une entité ne **peut pas** transmettre ses droits.

— **Role-Based Access Control** : Droits définis pour chaque catégorie de users, correspondant à un rôle spécifique.

3 Sujets, objets et droits d'accès

Un **sujet** est une entité pouvant accéder à des objets.
Trois catégories de sujets, **propriétaire**, **groupes d'utilisateurs** ayant des droits spécifiques, **reste du monde**.

Un **objet** est une ressource dont l'accès est contrôlé.

Un **droit d'accès** décrit la manière dont un sujet peut accéder à un objet, en lecture, écriture, execution, suppression, création, recherche, etc...

4 Contrôle d'accès aux fichiers UNIX

UNIX implémente le modèle **Discretionnary AC**.

4.1 Gestion des fichiers UNIX

Gestion des fichiers pas **inode**, contenant divers informations et notamment les permissions associées.

Les répertoires sont une liste de noms associés à des pointeurs vers d'autres inodes.

4.2 Gestion des utilisateurs UNIX

Dans la plupart des systèmes UNIX, un utilisateur est associé à :

— Un identifiant numérique **uID**
— Un groupe primaire
— Eventuellement à d'autres groupes, chacun identifié par un **group ID**

Un **user ID** de **superuser** existe, et possède un accès illimité à toutes les ressources systèmes.

4.3 Permissions d'accès aux fichiers UNIX

Quand un fichier est créé :

— Un propriétaire lui est associé et marqué du **userID**
— On lui associe un **groupID**, du groupe primaire du user

En plus, douze bits de protections sont associés au fichier, **9** de permission, **3** pour des comportements plus spécifiques :

— Les **3** premiers indiquent les permissions en lecture, écriture et exécution pour le **propriétaire**. Dans le cas d'un **répertoire**, le premier indique le listing, le second la création, le renommage et la suppression, et le troisième indique la recherche sur nom de fichier ou la descente dans le repertoire.

— Les **3** suivants indiquent la même chose pour le **groupe**.
— Les **3** suivants indiquent la même chose pour les **autres**.
— Le **10e** est le bit **setUID**, à 1 il indique que lorsqu'un utilisateur possédent les droits d'exécution l'exécute, il possède temporairement les droits du créateur (**effective userID**).
— Le **11e** est le bit **setGUID**, à 1 même comportement que le setUID mais pour un groupe. Pour un **répertoire**, indique que les fichiers créés seront associés au groupID du répertoire.
— Le **12e** est le **sticky bit**, indique pour un **répertoire** que seul le propriétaire peut renommer, déplacer ou supprimer un fichier.
Utilité pour fichiers partagés temporairement.

4.4 Utilisation d'Access Control Lists en UNIX

Si le système est composé de nombreux utilisateurs et de groupes, il est plus simple de configurer des **listes de contrôle d'accès (ACL)**, plutôt que d'associer les utilisateurs à plusieurs groupes, etc...

5 Types de mesures de contrôle d'accès

Politique de contrôle d'accès implique des mesures à différents niveaux et types :

— **Découragement** : Décourager attaquant via complexification des attaques potentielles
— **Prévention** : Anti-virus, etc...
— **Correction** : Rétablissement composants après incidents
— **Récupération** : Backups, etc...
— **Détection** : IDS
— **Compensatoire** : Mesures se couvrant les unes les autres
— **Directive** : Mesures obligatoires

6 Bonnes Pratiques

— Interdire accès aux utilisateurs non authentifiés
— Limiter et suivre usage des comptes particuliers (**admin**)
— Bloquer ou retarder accès après trop de tentatives infructueuses
— Retirer comptes des personnes licenciées
— Suspendre comptes inactifs
— Least Privilege Principle
— Désactiver services inutiles
— Remplacer les mots de passes par défaut
— Limiter et suivre les règles d'accès
— Forcer le changement régulier du mot de passe
— Protéger les fichiers de logs

7 Exercices

7.1 Questions de révision

— **Expliquez le modèle AAA**

Protocole de contrôle d'accès :

— **Authentication** : Vérification de la validité des éléments prouvant l'identité d'une personne ou d'une entité système.
— **Authorization** : Vérifier si la personne identifiée bénéficie des autorisations requises pour accéder au système.
— **Accounting** : Surveillance des accès aux ressources, logs, ...

— **Quelles sont les différences entre le modèle DAC, MAC, et RBAC ? Donnez un exemple d'utilisation**

— **Discretionary Access Control** : Contrôle d'accès basé sur des règles fixant quels demandeurs sont ou non autorisés à effectué une action, discretionnaire car une entité peut transmettre ses droits. (SQL)
— **Mandatory Access Control** : Contrôle d'accès basé sur le niveau de confidentialité (documents secrets, top secrets) qui est comparé aux droits d'accès du demandeur d'accès. Mandatoire car ces droits ne sont pas transmissibles.
— **Role Based Access Control** : Droits d'accès définis pour des catégories d'utilisateurs.

— **Quelle est la différence entre un sujet et un objet dans le contrôle d'accès ?**

— **Sujet :** Entité pouvant accéder à des objets.
— **Objet :** Ressource dont l'accès est contrôlé.

— **Listez quelques droits d'accès possibles**

  — Lecture
  — Ecriture
  — Exécution
  — Suppression
  — Création
  — Recherche

— **Expliquez que sont les bits Sticky, SetUID et SetGID dans le système de contrôle d'accès Unix classique.**

  — **Sticky Bit :** Indique que seul le propriétaire d'un fichier du répertoire peut renommer, déplacer ou supprimer ce fichier.
  — **SetGID :** Indique que les fichiers nouvellement créés seront associés au GID de ce répertoire.
  — **SetUID :** Lorsqu'un utilisateur possédant les droits d'exec sur un fichier l'exécute, on lui attribue temporairement les droits du user créateur du fichier.

— **Expliquez les droits d'accès Read, Write et Execute dans le cas d'un répertoire**

  — **Read :** Droit de listing
  — **Write :** Droit de création, renommage, suppresion de fichiers
  — **Execute :** Droits de recherche ou de traverser le dossier

— **En quoi est-ce que le bit SetUID peut éventuellement poser problème ?**

  Mettre un fichier et surtout un programme en Setuid ou Setgid n'est pas anodin car cela court-circuite le système de protection. Ainsi si vous tapez chmod ug+s /bin/bash vous donnez les droits root à toute personne qui ouvre un terminal ou qui lance l'interpréteur de commande bash. *Wikipédia : Setuid*

— **Expliquez brièvement ce que les Access Lists apportent par rapport au système de contrôle d'accès classique Unix**

  Le découpage des droits ne se fait plus sur les groupes mais sur les utilisateurs mêmes, une liste données donne tels drois pour tels fichiers à tels users.

— **Donnez et expliquez trois bonnes pratiques dans le cadre du contrôle d'accès**

  — Interdire l'accès aux utilisateurs non authentifiés
  — Désactiver les services inutiles
  — Retirer immédiatement les comptes des personnes qui quittent la société
  — Bloquer l'accès si trop d'essais infructueux

7.2 Questions de réflexion

1. **TODO**

Chapitre 6

Sécurité des DB & Cloud Computing

1 Sécurité des bases de données

Concentration d'informations **sensibles** voire **confidentielles**, nécessité pour l'entreprise de fournir un accès à ces données, mais également de les sécuriser.

En pratique, différence d'évolution entre les bases de données et leur sécurisation :

— Déséquilibre entre évolution des DB et techniques de sécurité
— Interactions via le SQL, plus complexe que HTTP et autres
— Pas de personnel dédié à la sécurité des DB en entreprise
— Environnements hybrides

2 Injections SQL

Exploitation de la nature des applications Web dynamiques, qui effectuent des requêtes SQL pour fournir ce contenu dynamique.

2.1 Principes

Principe : Véhiculer les requêtes SQL dans un traffic autorisé par le firewall. Ces requêtes sont insérées au milieu de requête normales générées par l'app Web avec l'input utilisateur.

Par exemple avec l'entrée suivante, où l'application attend le nom d'une ville de la part du user :

Liege' ; DROP table OrdersTable–

Le **' ;** termine le string et la commande SQL, le SGBD lancera donc après un DROP de la table OrdersTable, et tout le reste sera ignoré via le – qui marque le début de commentaires.

2.2 Points d'entrée des attaques

— **Input Utilisateur** : Formulaires et requêtes HTTP POST ou GET
— **Variables serveur** : En-têtes de protocoles
— **Injection de second ordre** : Stockage sur le système
— **Cookies**
— **Input physique** : Code-barre, formulaire papier

2.3 Types d'attaques

Trois types d'attaques : **inband**, par **inférence** et **outband**.

Attaques Inband

Utilisent le **même canal** pour l'injection et pour la récupération des résultats.
Ces attaques regroupent :

— **Tautologies** : Injecte une condition toujours **vraie** pour passer outre une vérification, typiquement **‘ OR 1=1 − ’**.

— **Commentaires de fin de ligne** : Insère le marqueur − pour annuler le reste du code SQL légitime.

— **Requêtes "PiggyBackées"** : Injecte une nouvelle requête au milieu d'un requête légitime.

Attaques par Inférence

Pas de transfert de données, observation des **réactions** du système.

— **Requêtes illégales** : Tentative d'obtenir des messages d'erreurs un peu trop descriptifs.
— **Injections SQL à l'aveugle** : Insertion de requêtes true/false pour observer la réaction du serveur.

Attaques Outband

Données récupérées via un autre canal comme le **mail**.

2.4 Contre-Mesures

Classifiables en trois catégories :

1. **Code Défensif**
2. **Détection**
3. **Prévention lors de l'exécution**

3 Contrôle d'accès aux bases de données

Contrôle d'accès par **rôle** (RBAC), ou **discretionnaire** (DAC).
Supportent plusieurs types de politiques administratives :

— **Administration centralisée** : Petit nombre de users privilégiés attribuent ou révoquent les droits d'accès
— **Administration par le propriétaire** : Propriétaire d'une table gère les accès à la table
— **Administration décentralisée** : Le propriétaire peut gérer les accès et déléguer ses droits

3.1 Définition des accès en SQL

SQL permet via **GRANT** et **REVOKE**, d'attribuer ou révoquer des droits selon certaines options.

L'option **GRANT OPTION** permet d'attribuer le droit de donner des droits, ce qui crée un modèle complexe en **cascade**.
La règle convention en cas de révoquation est que, si A révoque un droit d'accès, tous les droits qui

découlent sont révoqués, sauf s'ils avaient éxisté malgré tout sans le droit attribué par A.

3.2 Contrôle d'accès par rôle

Convient bien aux bases de données, pas évident d'attribuer des droits d'accès au cas par cas .

Trois catégories d'utilisateurs :

1. Propriétaire de l'application, possède les objets de la BDD
2. Utilisateur final, opère sur les objets à travers l'application
3. Administrateur, reponsable d'une partie ou de toute la BDD

Une base de données RBAC doit fournir les fonctionnalités suivantes :

— Création et suppression de rôles
— Définition d'une permission pour un rôle
— Assignation et révocation d'un rôle à un ou plusieurs utilisateurs

4 Inférence

Effectuer des opérations **autorisées** et d'en déduire les informations non accessibles.

Deux techniques :

— Analyser les dépendances fonctionnelles
— Regrouper des vues ayant les mêmes contraintes

Deux approches pour gérer cette menace :

1. **Détection d'inférence à la conception** : Altère la structure ou change le contrôle d'accès pour éviter l'inférence
2. **Détection d'inférence à la requête** : Rejet des requêtes détectées comme inférentes

5 Chiffrement de la base de données

Deux désavantages au chiffrement des bases de données :

— **Gestion des clés** : Les utilisateurs doivent disposer de la clé de déchiffrement, gestion sécurisée complexe si users nombreux
— **Inflexibilité** : Recherche complexifiée dans la base de donnée

6 Cloud Computing

Cloud Computing : Modèle permettant un accès réseau pratique, à la demande et omniprésent à un pool partagé de ressources informatiques, qui peut être rapidement provisionné et déployé avec un effort de gestion ou une interaction avec le fournisseur minimal.

Cinq caractéristiques essentielles :

1. **Accès réseau large** : Fonctionnalités accessibles par des mécanismes standards
2. **Elasticité rapide** : Permet d'ajouter ou retirer des ressources selon besoin

3. **Service mesuré** : Utilisation de chaque service monitorée, contrôlée, reportée
4. **Self-service à la demande** : Ajustement auto des ressources par le client
5. **Pooling de ressources** : Ensemble des ressources regroupées pour réattribution facile

Trois modèles de service :

1. **SaaS** : Application logicielle cloud, webmails
2. **PaaS** : Plateforme dans le cloud, GoogleAppEngine
3. **IaaS** : Infrastructure, machines, OS virtualisés ans le cloud

Quatre modèles de déploiement, différencie le niveau de responabilité selon le fournisseur ou organisation :

1. Cloud publique
2. Cloud privé
3. Cloud communautaire
4. Cloud Hybride

7 Risques de sécurité et contre-mesures liées au Cloud Computing

Risques supplémentaires car l'organisation perd une partie du contrôle sur les ressources, services et applications.
Elle doit assuer la **tracabilité** au niveau des politiques de sécurité et de confidentialité.

Menaces principales :

— Abus et usage malicieux du Cloud Computing (BotNet DoS, Spam)
— Interfaces et API non sécurisées
— Attaques internes, personnel du fournisseur Cloud
— Problèmes de technologies partagées
— Pertes ou fuites de données
— Piratage des comptes ou services

8 Cloud Security As A Service

Services de sécurisation fournis par les fournisseurs Cloud, comprenant **authentification**, **antivirus**, **IDS**.

Systèmes de Backups pour les **Disaster Recovery**.

9 Exercices

9.1 Questions de révision

— **Expliquez le concept d'autorisation en cascade**
Si Alice donne à Bob ses droits, Bob peut également les donner à Charlie, qui peut ensuite en donner à Daryl.
Si maintenant on décide de retirer les droits accordés à Bob par Alice, ce retrait va se faire en cascade et Charlie et Daryl vont perdre les droits qu'Alice avait légués par Bob et que lui-même leur avait légué.

— **Expliquez l'injection SQL par tautologie**
On fait en sorte de modifier la requête pour y injecter une condition toujours vraie pour passer outre la vérification de mots de passe.

— **Expliquez la menace d'inférence sur une base de données**
L'attaquant observe la manière dont le système réagit à l'attaque pour en déduire des informations sur son comportement, en essayant par exemple d'afficher des messages d'erreurs trop descriptifs, ou insérant des requêtes SQL à l'aveugle.

— **Quels sont les désavantages du chiffrement d'une base de données ?**
Gestion des clés complexes, complexifie et ralenti le process de la base de données (qu'on veut rapide la plupart du temps).
— **Listez et décrivez brièvement les trois modèles du service Cloud**
  — **Software As A Service :** Application logicielle disponible dans le cloud, telle que le client mail GMail
  — **Platform As A Service :** Plateforme sur laquelle une organisation peut développer et faire tourer ses propres applications, telle Google App Engine.
  — **Infrastructure As A Service :** La ressource est matérielle (machines + OS) contrôlables via une API, tels Windows Azure, DigitalOcean.

— **Décrivez trois menaces de sécurité spécifiques au cloud**
  — Abus et usage malicieux du Cloud, création d'un botnet de machines dans le cloud.
  — Interfaces et API non sécurisées.
  — Pertes ou fuites des données.

9.2 Questions de réflexion

1. **TODO**

Chapitre 7

Malwares

Malicious Softwares : Programme inséré dans un système avec l'intention de compromettre la CIA des données, application, OS ou autre ressource de la victime.

1 Types de Malwares

— **Advanced Persistent Threat** : Cybercrime dirigé sur une cible business ou politique, large panel de méthodes et sur de longues durée, souvent menée par des organisations soutenues par des états
— **Adware** : Publicité intégrée dans un logiciel, redirections ou popup intempestifs
— **Attack Kit** : Pannel d'outils générant de nouveau malwares automatiquements
— **Auto-Rooter** : Outil utilisé pour s'introduire dans de nouvelles machines à distance
— **Backdoors** : Mécanismes bypassant les vérifications de sécurités
— **Downloaders** : Code installant d'autres éléments sur la machine visée
— **Drive-by Downloaders** : Exploite une vulnérabilité browser pour attaquer le client
— **Exploits** : Code spécifique à une vulnérabilité particulière
— **Flooders (DoS client)** : Génère un gros volume de données pour une attaque par botnet
— **Keyloggers** : Capture des frappes au clavier
— **Logic Bomb** : Code malicieux destiné à se déclancher sous certaines conditions
— **Macro Virus** : Virus à base de scripts ou macros
— **Mobile Code** : Code pouvant tourner sur de nombreuses plateformes
— **Rootkit** : Outils de hackers destinés à s'introduire sur un système et gagner les accès root
— **Spammer Programs** : Envoi de larges quantités d'emails indésirables
— **Spyware** : Collecte des infos sur un ordinateur et les envoie sur un autre système
— **Trojan Horse** : Programme malicieux se faisant passer pour bénéfique
— **Virus** : Malware se réplicant à l'exécution
— **Worm** : Programme tournant indépendamment et se propageant sur les hôtes et le réseau
— **Zombie, bot** : Programme tournant sur une machine infectée, destiné à attaquer une autre cible.

2 Mode de propagation

Trois types de propagation :

1. Contenu infecté (virus)
2. Exploitation de vulnérabilités (vers)
3. Techniques de social engineering (trojan et spam)

2.1 Virus

Morceau de programme pouvant infecter d'autres programmes ou executables.
Analogie avec el virus biologique et sa **réplication** dans l'hôte.

Le virus informatique possède des instructions dans son code destiné à sa réplication et à sa propagation de proche en proche, ou dès qu'il entre en contact avec un nouvel hôte (USB, Mail).

Ils sont composés de trois parties :

1. Vecteur ou mécanisme d'infection
2. Element déclancheur (logic bomb)
3. Action (destruction de données)

Quatre phases dans leur cycle de vie :

1. Phase dormante
2. Phase de propagation
3. Phase de déclanchement
4. Phase d'exécution

2.2 Vers

Programme cherchant de manière active de nouvelles machines à infecter.
Ils exploitent des **vulnérabilités software** afin de gagner l'accès à de nouveau systèmes, et peuvent utiliser les médias partagés, réseau ou scripts et macros.

Comme les virus, les vers ont une phase de propagation et plusieurs actions associées.

Moyens de réplication :

— Emails, pièces jointes
— Partage de fichiers
— Execution à distance
— Transfert ou accès à dinstance aux fichiers
— Connexion à distance

Découvertes de nouvelles cibles :

— Scan aléatoire du réseau
— Hit-List, liste de machines potentiellement vulnérables
— Scan topologique
— Scan du réseau local

2.3 Spam et Trojan

Propagation en trompant les users pour qu'ils se compromettent eux-même et leur système sans le savoir.

Spam

Evalué à 90% du traffic mail total, lutte technologique entre spammers et anti-spams.

Croissance stabilisée, les réseaux sociaux sont un nouveau moyen de spammer.

Origine de BotNets.

Simple publicité ou tentatives d'escroquerie, voire vecteurs de propagations de malwares, phishing.

Trojan

Cheval de troie : Programme utile en apparence contenant des instructions cachées qui effectuent des actions nocives.

Utilisés pour accomplir des actions indirectement comme création de backdoors, accès à des infos personnelles, scan de fichiers et copie via mail ou web.

3 Types d'actions

3.1 Corruption des systèmes

— Destruction des données, ou chantage par menace de destruction ou chiffrement des données
— Dommages physiques, modification de contrôle de processus industriel

3.2 Agent d'attaque

Machine détournée et utilisée comme agent (bot) d'un **botnet**, typiquement utilisés pour :

1. DDoS
2. Spam
3. Sniffer le traffic
4. Keylogging
5. Automatisation de clics publicitaires
6. Manipulation de jeux ou paris en ligne

3.3 Vol d'information

Vol de logins sur ebanking, de jeu, ou même de documents techniques ou informations de configuration à des fins d'espionnage.

Keylogger, spywares ou phishing.

3.4 Furtivité

Typiquement des **backdoors**, point d'entrée dissimulé sur un système.

Rootkit, couverture d'accès en root illicite, obtention de backdoors à la demande.

4 Les Contre-Mesures

Connues sous le nom d'**anti-virus**.
Principale arme = **prévention** et **sensibilisation** des utilisateurs.
Si la prévention échoue :

1. Détecter l'infection et localiser le malware
2. Identifier le malware
3. Retirer le malware et toute trace de son passage

5 Exercices

5.1 Questions de révision

— **Quels sont les trois modes de propagation de malware ? :**

  — Via du contenu infecté (**virus**).
  — Via l'exploitation de vulnérabilités (**vers**).
  — Via techniques de social engineering (**spams et trojan**).

— **Catégories d'actions qu'un malware peut effectuer :**

  1. Corruption des systèmes
  2. Agents d'attaque
  3. Vol d'information
  4. Furtivité

— **Quelles sont les phases typiques du cycle de vie d'un virus**

  1. Sommeil
  2. Propagation
  3. Déclanchement
  4. Exécution

— **Quel moyen un ver peut-il utiliser pour accéder à des systèmes distants pour se propager ? :**
  — Scan aléatoire
  — Hit-list
  — Scan topologique
  — Réseau local

Chapitre 8

Attaques par Déni de Service

1 Généralités

Les **DoS** et plus particulièrement les Distributed Denial of Service **DDoS** existent depuis longtemps.

DoS : Action qui empêche ou complique l'usage légitime de réseaux, systèmes ou applications en saturant les ressources telles que le CPU, la mémoire, la bande passante ou l'espace disque.

Trois catégories de ressources ciblées :

— **Bande-passante** : Saturation des liens avec énorme traffic
— **Ressources système** : Paquets destinées à consommer les ressources système, ping of death, SYN flooding
— **Ressources applicatives** : HTTP flood, cyberslam avec requêtes SQL coûteuses

1.1 Attaques DoS classiques

Volonté de surcharger la capacité réseau de la cible, la bande-passante de l'attaquant doit donc être plus large que celle de la cible.
Envoi massif de ping vers la cible, consommant les ressources du lien et empêchant le traffic légitime de passer.

Faiblesses : origine de l'attaquant retrouvée via son ip, et les ping icmp renvoie une réponse, ce qui dégrade les ressources du réseau de départ.
Améliotions : Spoofing de l'adresse source via forging des paquets IP, les réponses seront renvoyées vers les adresses usurpées, qui pourront renvoyer des messages d'erreur à la cible et accroître l'attaque.
Solution : Filtrage des IP par les FAI pour éviter le forging, peu ou pas mis en place par les FAI.

Autre type d'attaque via **SYN Spoofing**, nécéssite beaucoup moins de volume que le ping flood :

— Attaquant génère des demandes de connexion TCP SYN avec une ip source spoofée
— La cible enregistre les demandes et renvoie des SYN+ACK
— Si la source spoofée est valide, elle renverra des RST
— Sinon, le serveur gardera des demandes de connexions ouvertes dans sa table TCP, **occupation de la table** et injoignabilité du serveur visé

2 Attaques par flooding

Caractérisées par protocole réseau utilisé, but de congestion et de rendre indisponible le serveur cible.

2.1 ICMP Flood

ICMP Flooding classiquement par echo request, mais généralement bloqués par les firewalls de nos jours.

Certains sont cependants nécéssaires et non bloqués tels que **time exceeded** ou **destination unreachable**.

2.2 UDP Flood

Envoie massif de segments UDP sur un port avec par exemple le service **diagnostic echo**.

2.3 TCP SYN Flood

A la différence du SYN Spoofing, le **SYN FLooding** consiste à saturer la bande passante de demandes de connexions TCP.

Limité par le volume de données que la source peut générer, et la présence ou non de spoofing ou de botnet.

3 DDoS

L'attaquant possède un **botnet**, réseau de machines **zombies** infectées par un malware et qui participeront à l'attaque.

Contrôle des bots via communication chiffrée pour éviter de laisser des traces.

4 DoS Applicatives

But d'impact un **système** et non un lien réseau, ce en forçant le système à effectuer des opérations coûteuses.
On vise une application spécifique en utilisant une **faible** bande passante.

4.1 SIP Flood

Flux de requêtes **INVITE** sur un Proxy, qui va devoir utiliser DNS et transferer les requêtes pour chacune, et sera donc surchargé d'une part par la bande passante, d'autre par le traitement CPU à effectuer.

Destination également impactée par la réception de nombreux appels simultanés.

4.2 Attaques HTTP

HTTP flood

Bombardement d'un serveur Web de requêtes HTTP, demandant par exemple un fichier de grosse taille. Ceci va consommer du CPU et de la bande passante.

Variante appelée **recursive HTTP flood** ou **spidering**, consite à envoyer une requête sur un lien et suivre tous les liens de la page récursivement.

4.3 Slowloris

Occupation des ressources d'un serveur Web via des requêtes HTTP **partielles** qui vont occuper les sockets serveur.

Une fois tout les sockets TCP et thread associés monopolisés, les users ne peuvent plus accéder au serveur. Slowloris continue d'envoyer à intervalles régulier ces requêtes, afin de garder la connexion HTTP ouverte.

5 Attaques par réflexion et amplification

Variante de DDoS où les relais sont des systèmes sains et pas des botnets.
Attaque via spoofing de l'adresse source, qui est la cible.

Envoi de paquets à une série de serveurs qui vont répondre vers l'adresse spoofée et saturer sa bande-passante.

Simples à déployer car utilisent des mécanismes réseau et systèmes **légitimes**.

5.1 Attaques par réflexion

Basée sur des relais intermédiaires, choisis pour leur large bande passante, afin de noyer la victime dans un traffic régulier.

Service choisi pour sa capcité à générer des tailles de réponses plus larges que les requêtes.
Services UDP classiques utilisables, DNS ou SNMP, réponses jusqu'à 512 octets pour requête de 60 octets.
Variante avec paquets TCP SYN, difficiles à détecter puisque d'apparences légitime.

Unique solution via ISP et filtrage des ip spoofées.

5.2 Attaques par amplification

Message initial déclenchant de multiples paquets réponses vers l'adresse spoofée, par exemple via broadcast.
Difficulté de trouver un service utilisé par tout les hôtes du réseau broadcast, **echo request** est un bon exemple.
Smurf et **fraggle** en sont des exemples.

Prévention en bloquant les messages de broadcast initiés à l'extérieur du réseau de destination, limitation des pings et echo.

6 Protection contre les attaques DoS

Quatres lignes de défenses contre les **DoS** :

1. **Prévention et anticipation** : Difficile de prévenir, filtrage des IP spoofée par les ISP. Anticipation vise à mettre en place des mécanismes de survie en cas de DoS, ressources de réserve, backups.

2. **Détection et filtrage** : Détection le plus tôt possible via IDS, filtrage des paquets concernés, collaboration avec les ISP via un autre medium.

3. **Traçage et identification de la source** : Eviter des attaques ultérieures en remontant à la source de l'attaque.

4. **Réaction à l'attaque** : Restauration du système et réparation des dégats effectués.

7 Exercices

8 Questions de révision

— **Quels types de ressources sont visés par les attaques DoS ? :**

  — Bande-passant réseau
  — Ressources système
  — Ressources applicatives

— **Quel est le but d'une attaque DoS ? :**

  Empêcher ou compliquer l'usage légitime de réseaux, systèmes ou applications en saturant les ressources.

— **Quels types de paquets sont utilisés pour les attaques DoS ? :**

  — ICMP
  — TCP
  — UDP
  — HTTP

— **Pourquoi est-ce que beaucoup d'attaques DoS utilisent le spoofing d'adresses ? :**

  Pour ne pas que l'on puisse remonter à la source de ces attaques.

— **Quelles sont les défenses possibles contre les attaques TCP SYN flooding ? :**

  Identifier l'attaquant.

— **Que faut-il faire lorsqu'une attaque DoS est détectée ? :**

  Filtrer les paquets concernés, collaborer avec le FAI, traçer la source et l'indentifier.

— **Qu'est-ce qu'une attaque par réflexion ? Et par amplification ? :**

  — Réflexion : Se base sur des relais intermédiaires pour rediriger un flux vers la cible.
  — Amplification : Rediriger des paquets amplifiés (réponses DNS) sur une IP spoofée.

Chapitre 9

Détection d'intrusion

Intrusion : Evénement(s) en sécurité qui constituent un incident au cours duqel un intrus gagne ou essaie de gagner l'accès à un système sans en avoir l'autorisation.

Par exemple : vol de login, acquisition de privilèges non autorisés, malwares, etc...

1 Les Intrus

Hackers ou crackers cherchant à s'introduire dans un système, cible aléatoire ou spécifiques après investigation.
Attaques conçues pour passer outre les firewalls et autres défenses.

La motivation du cracker définit la stratégie à appliquer :

— **Cyber-Criminels** : Objectifs de gains financiers, vols de données
— **Activistes** : Motivés par des causes politiques ou sociales, DoS, nuisance à la réputation
— **Organisations Sponsorisée par l'état** : Attaques d'espionnage ou de sabotage, destruction
— **Autres** : Hackers classiques, motivés par le challenge ou leur réputation
Différents niveaux d'expertises :

1. **Apprentis** : Niveau technique minimal, **script-kiddies**
2. **Compagnons** : Niveau technique suffisant, capable de modifier et étendre des toolkits
3. **Maîtres** : Niveau technique élevé, capables de découvrir de nouvelles failles

Selon leur expertise et leur motivation les intrusions peuvent être de bénignes à sérieuses.

Les **IDS** sont conàus pour aider à contrer ces menaces, mais ne pourront pas faire grand chose contre les attaques complexes ou exploit de vulnérabilité **zero-day**.

2 Etapes d'une intrusion

1. **Choix cible et collecte d'information** : Information publique sur le système, exploration du site web, port scanning.
2. **Accès initial** : Utilisation d'une vulnérabilité du réseau, crackage du mot de passe, malware.
3. **Acquisition de privilèges** : Via vulnérabilité logicielle ou sniffer pour capturer les motsde passes
4. **Collecte d'info ou exploitation du système** : Scan de fichiers, obtention de mots de passes vers d'autres machines
5. **Pérénnisation de l'accès** : Backdoors, malwares ou crédences d'authentification
6. **Dissimulation des traces** : Permet à l'attaquant de masquer son passage

3 La détection d'intrusion

Intrusion Detection : Service de sécurité qui surveille et analyse les events système dans le but de trouver des tentatives d'accès non autorisées aux ressources du système, et de fournir des avertissements en temps réel ou presque.

Trois composantes logiques :

1. **Senseurs** : Collecte de données (paquets, logs, appels système)
2. **Analyseur** : Recoit données de senseurs et les analyse, indique si intrusion
3. **UI** : Permet au user de voir les résultats de l'analyse

Nombre de senseurs définit le type d'IDS :

— **Host-Based IDS** : Surveille hôte unique
— **Network-Based IDS** : Surveille trafic réseau
— **IDS Hybrid** : A la fois HIDS et NIDS

Les objectifs d'un IDS sont de **détecter les intrusion** le plus vite pour limiter les dégats, et **identifier** les attaquants pour les éjecter.
Ensuite, un IDS efficace **décourage** les intrusions. Enfin, permet la collecte d'infos pour renforcer la prévention.

L'IDS essaie de détecter les comportements anormaux, risque de **faux-positif**, ainsi que de **faux-négatifs**, nécéssite un bon ajustement !

4 Approches analytiques

IDS exploitent les senseurs de deux manières :

1. **Recherche d'anomalie** : Collecte des comportements légitimes pour comparaison
2. **Application heuristique (découverte) ou reconnaissance de signatures** : Base de données comportant la signature des intrusions

5 Host-Based IDS

Détecte à la fois les attaques internes **et** externes.

Senseurs utilisés :

— Traces d'appels-systèmes
— Fichiers de logs
— Checksum d'intégrité des fichiers
— Accès au registre

Utilisent la détection d'**anomalies** et de **signatures**.

6 Network-Based IDS

Le **NIDS** surveille le trafic à des endroits spécifiques du réseau, implantés dans le pérmiètre de sécurité.

Senseurs actifs, ou passifs qui analyse des **copies** des paquets pour ne pas impacter le délai.

Localisation à différents endroits :

1. **Firewall Externe** : Voit attaques de l'extérieur qui ont pu pénétrer
2. **Extérieur du réseau** : Surveille trafic entier non filtré
3. **Backbone du réseau donnant accès aux services principaux** : Visibilité sur activités de l'intérieur
4. **LAN ou machines spécifiques** : Surveille intrusions sur système critiques

Détection d'**anomalie**, sur base des patterns de trafic. **Signature** pour les attaques spécifiques telles que ICMP ou Ip spoofées, ou encore SYN floods.

7 IDS Hybrides

Ensemble de senseur répartis au niveau des hôtes et à divers endroits du réseau.
Analyseur centralisé, vue d'ensemble de l'organisation.

8 HoneyPots

Leurre, machine **sans valeur productive** conçue pour attirer les attaques :

— Distrait l'attaquant
— Collecte de l'information sur l'activité de l'attaquant
— Encourage l'attaquant à rester sur le système assez longtemps pour que l'admin réagisse

Honeypots à **faible interaction** sont des logiciels imitant des services, ceux à **forte** interaction sont des services réels déployés dans le but de Honeypot.

9 Exercices

9.1 Questions de révision

1. **Listez et définissez quatres types d'intrus :**

   (a) **Cyber-Criminels :** Objectif financier, via vol d'indentité, accès à des organismes financiers, espionnage, vol de données.

   (b) **Activistes :** Causes politiques ou sociales, veulent promouvoir leurs causes via Defaces ou DoS Attacks.

   (c) **Organisations sponsorisées par un état :** Advanced Persistant Threats, attaques d'espionnage ou de sabotage.

   (d) **Autres :** Hackers classiques motivés par le challenge ou la reconnaissances de leurs pairs.

2. **Donnez les principales étapes d'une intrusion avec un exemple d'activité :**

   — **Choix de la cible** : Scanning NMAP, WhoIS
   — **Accès initial** : Crackage de mot de passe, MalWare, Social Engineering
   — **Acquisition de privilèges** : Vulnérabilités logicielles ou sniffer pour obtenir un mot de passe

— **Collecte d'informations ou exploitation du système** : Scan de fichiers, obtention de mots de passes vers d'autres réseaux.
— **Pérénnisation de l'accès** : Installation de Backdoors
— **Dissimulation des traces** : Rootkits, modification des logs

3. **Quelle est la différence entre un HIDS et un NIDS ? Comment peut-on les combiner ?**

   Un HIDS surveille un hôte uniquement sur base des events qui se produisent dessus, un NIDS surveille le trafic réseau de certains segments ou appareils spécifiques.
   On peut les combiner sur in IDS Hybride qui combine ces différents senseurs.

4. **Quel sera l'impact d'un faux positif avec un NIDS ? Et un faux négatif ? :**

   Trop de faux positifs amènera l'équipe d'administration à les ignorer, un faux négatif sera plus pernicieux car considéré comme légitime et de base moins surveillé qu'un faux positif.

5. **Quels types de senseurs peut utiliser un HIDS ? Et un NIDS ?**

   Pour un HIDS :

   — Traces d'appels systèmes
   — Fichiers de logs
   — Checksums d'intégrité des fichiers
   — Accès au registre

   Pour un NIDS :

   — Détection d'anomalie
   — Détection par signature

6. **Où placer un NIDS ?**

   — Niveau du firewall externe
   — Extérieur du réseau
   — Au niveau du backbone du réseau
   — Au niveau du LAN et machines spécifiques

7. **Qu'est-ce qu'un honeypot ?**

   Un leurre conçu spécifiquement pour attirer les auteurs d'attaques et les distraire, collecter des infos sur ces attaques et les retenir assez longtemps pour permettre une réaction de l'administrateur.

Chapitre 10

Les Firewalls

1 Motivations

Actuellement plus possible d'exposer son réseau, il faut protéger l'ensemble de celui-ci via un moyen de **contrôler** et **réguler** tout le trafic entrant.
Ceci est réalisé par un firewall, forçant tout le trafic à passer par lui, point central pour politiques de **sécurité**, **surveillance** et **audit**.

2 Caractéristiques et politiques d'accès

Objectifs des firewalls :

1. Tout le trafic entre extérieur et intérieur doit traverser le firewall
2. Seul le trafic autorisé doit passer
3. Firewall protégé contre intrusions

Politiques d'accès définies sur base de l'**analyse des risques** par l'organisation.

Firewall outil indispensable pour empêcher les intrusions et simplifier la gestion de la sécurité, peut héberger des services complémentaires (IPS, NAT, IPSec).

3 Types de firewalls

3.1 Firewalls stateless

Analyse chaque paquet sur base d'un **ensemble** de règles, qui dit si le paquet doit être transmis ou jeté.

Des policies, de **DROP** ou **ACCEPT** sont mises en places pour un paquet qui ne rencontrerait aucune des règles, **DROP** est bien entendu préféré.

Les firewalls **stateless** souffrent cependant de certaines faiblesses, ils ne peuvent pas repérer les attaques employant des vulnérabilités **applicatives**.

Ils sont eux-mêmes vulnérables à certaines attaques :

— Spoofing IP
— Attaque par source routing
— Attaque par fragmentation

3.2 Firewalls stateful

Filtrage plus spécifique sur analyse du contexte de la connexion TCP, garde en mémoire les informations de connexions.

3.3 Proxies

Relais au niveau de la couche applicative, le proxy se positionne entre un hôte à l'intérieur du réseau protégé et son interlocuteur à l'extérieur.

Le proxy intercepte les requêtes pour une application donnée, effectue le **filtrage**, l'**authentification**, et relaie enventuellement la requête à son destinataire final.

Permet de selectionner le type de trafic accepté, ainsi que de faire de la surveillance et des logs au niveau applicatif.

Introduisent cependant un délai supplémentaire.

Proxy WEB/HTTP

Intercepte les requêtes HTTP des machines internes, peut effectuer certaines opérations :

— Authentification user
— Filtre sur contenu (porn, jeux)
— Cache
— Compression
— Anonymisation
— Surveillance et log

Egalement utilisé pour :

— Protéger les utilisateurs
— Contourner des politiques de sécurité
— Accéder à du contenu limité à certains pays
— Requêtes anonymes

Reverse Proxies Web

Le reverse proxy protège le **serveur**, rempart évitant l'accès direct au serveur web.
Optimise l'accès aux ressources HTTP, protège l'intérieur du réseau.
Typiquement placé en **DMZ**.

Opérations lors du relai :

— Cache pour alléger la charge du serveur
— Décharger le serveur en servant de terminateur SSL (déchiffre les connexions)
— Sécurisation en centralisant les connexions
— Optimiser performances via compression
— Load-Balancing sur plusieurs serveurs web

Proxies Mail

Utilisés pour détecter de bloquer les attaques **spam**.

Proxies FTP

Filtrage et authentification ou limite dans commandes FTP pour reverse proxy.

Proxies DNS

Filtrage des requêtes non-conformes ou présentant un risque sécuritaire.

3.4 Passerelles niveau circuit

Proxy au milieu de la connexion TCP, créant deux connexions séparées, filtrage au niveau transport.

Utilisé en environnement **fiable**.

4 Types d'installation d'un firewall

4.1 Bastion

Point de passage **critique** dans le réseau, typiquement point entrée/sortie avec Internet.
Set aussi à héberger les proxies ou passerelles niveau circuit.

Caractéristiques suivantes :

— Machine entièrement sûre, hardening, services minimals et light
— Mécanismes d'authentification avant accès aux services
— Proxies configurés pour fournir uniquement certain services pour chaque application
— Logging
— Proxies indépendants les uns des autres
— Applications des proxies sans disque, en read-only, pas d'installation possible de sniffers ou trojans

4.2 Firewall sur hôte

Sécurisation de machines individuelles, permet de :

— Ajuster règles firewall à l'environnement hôte
— Protection indépendante de la topologie et de l'emplacement de l'hôte

Si la machine est un serveur, firewall configuré en fonction du service spécifique.
Si machine user, firewall plus **généraliste**, emoêche accès extérieurs à la machine et filtre et détecte le trafic émis par un ver ou malware.

5 Localisation et configuration du firewall

Destiné à faire **barrière** entre une source de trafic extérieur et un réseau interne.

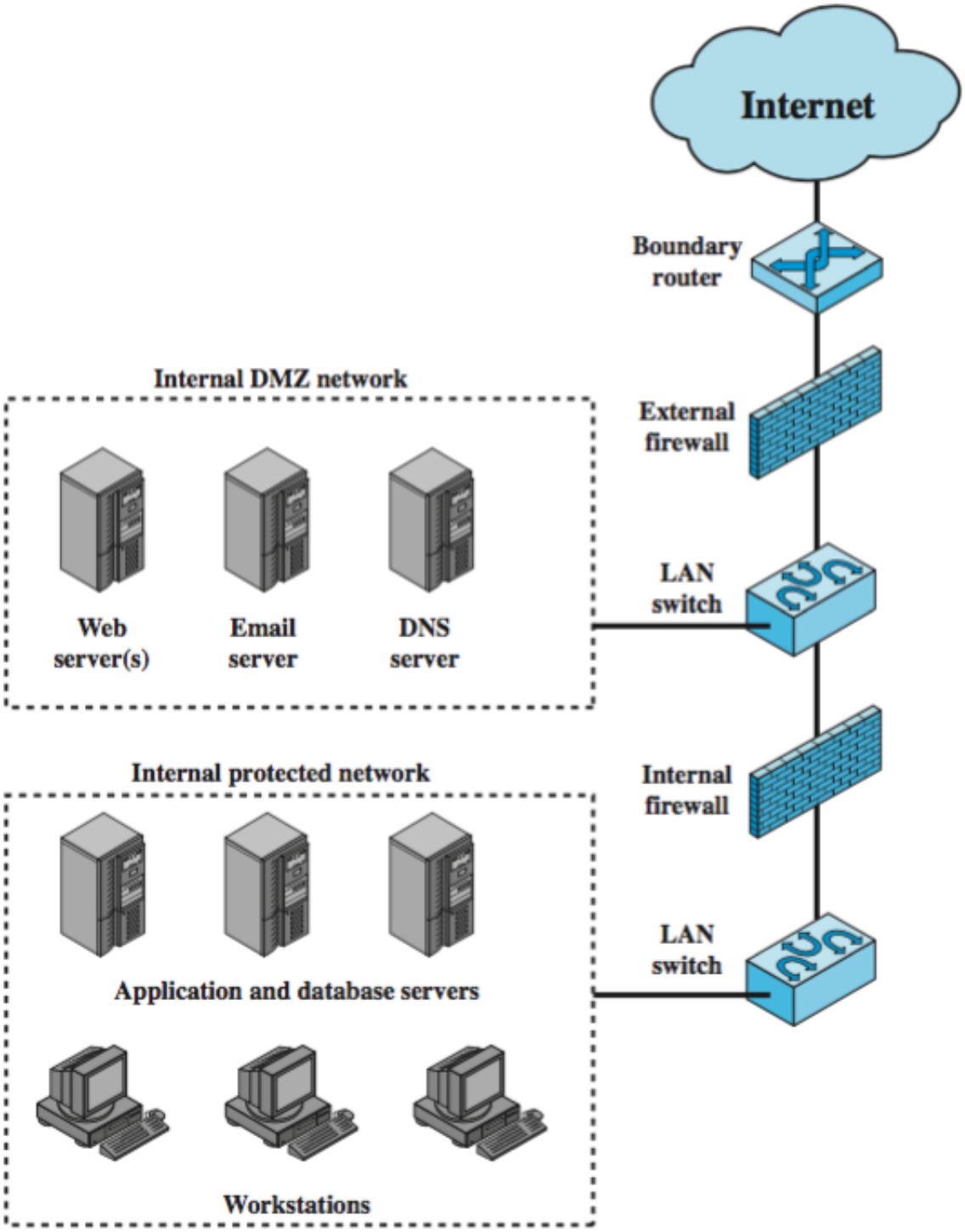
Cela laisse plusieurs choix pour le placement de ces firewalls :

5.1 Réseau DMZ

Zone tampon avec les services qui doivent êtres accessibles de l'extérieur mais nécéssitant une protection, typiquement entre deux firewalls.

Le firewall externe protège l'accès à la DMZ, ceux internes auront trois objectifs :

1. Filtrage plus restrictif que le firewall externe pour protéger les serveurs et postes de travail des attaques extérieures
2. Protection au réseau interne contre attaques depuis la DMZ, et protège la DMZ contre des attaques depuis le réseau interne
3. Protection de portions du réseau interne



5.2 VPN's

Les **VPN** permettent d'interconnecter des machines à travers des réseaux non sécurisés, via utilisation de protocoles de sécurisation et couche de chiffrement.
Essentiel pour éviter espionnage et protéger l'accès au réseau.

Deux scénarios pour les VPN :

— Interconnexion de deux sites distants d'une même entreprise
— Permettre aux utilisateurs nomades d'accéder au réseau d'entreprise

VPN utilisant IPSec, les firewall servent de terminaisons IPSec en bordure du réseau afin de gérer l'encapsulation, décapsulation du traffic IPSec.

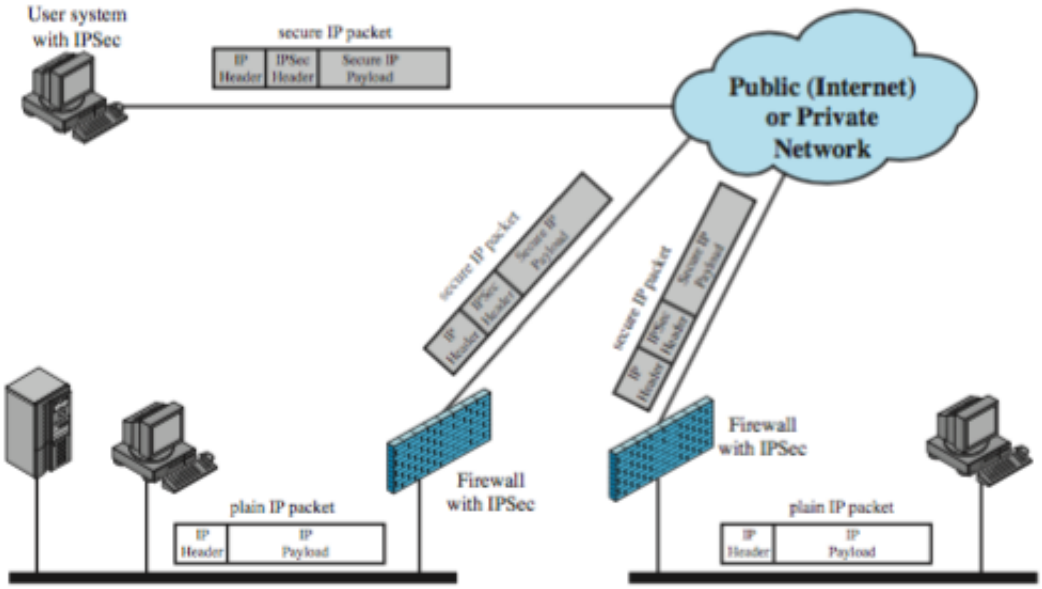


Figure 9.4 A VPN Security Scenario

5.3 Firewalls distribués

Combinaison des types de firewalls (stand-alone et host) pour former un système de **firewall distribué**.
Permettent une configuration des politiques de manière centralisée, collecte et consultation des logs doit être soignée pour surveiller l'entièreté du réseau.

L'image suivante montre une configuration de ce type :

— DMZ extérieure, configurée spécifiquement pour le web
— Firewall extérieur protège l'ensemble du réseau (**stand-alone**)
— Seconde DMZ interne, serveurs accessible de l'extérieur de manière très contrôlée, chacun à son propre firewall
— Firewall interne isole la partie la plus vulnérable du réseau, chaque machine est aussi protégée par un **host firewall**.

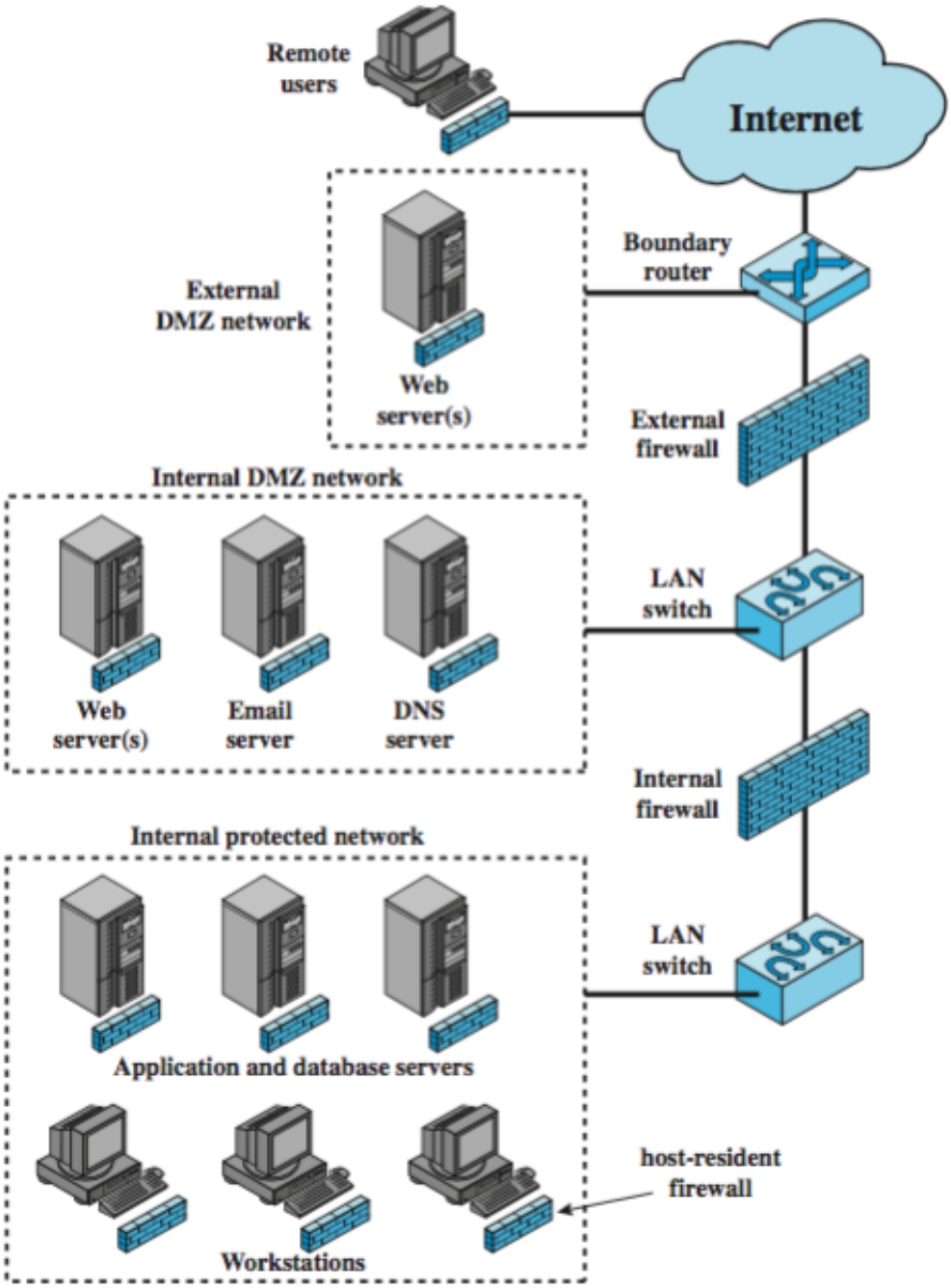


Figure 9.5 Example Distributed Firewall Configuration

5.4 Synthèse

— **Firewall-Hôtes** : Firewall personnels et spécifiques au serveur
— **Routers-Firewall** : Routeurs placés entré deux réseaux avec capacités de filtrage
— **Bastion ligne simple** : Firewqall placé entre routeur interne et externe, capture le traffic entre ces routeurs
— **Bastion simple T** : Utilise **3** interfaces pour contrôler l'accès à un trosième réseau type DMZ
— **Deux bastions en ligne** : Deux firewalls bastions encadrent la DMZ, protection de l'extérieur ET de la DMZ si compromise. Redondance physique.

— **Deux bastions en T** : Firewall externe connecte la DMZ sur une interface séparée de celle le

connectant au firewall interne.

— **Firewall Distribués**

6 Systèmes de prévention d'intrusion

Intrusion Prevention System, fournissent une détection de comportements suspects sur base de la signature. Deux types :

— **Host-Based IPS** : Installé sur une machine spécifique, se concentre sur le traffic généré par les applications et séquences d'appels-systèmes pour tenter d'identifier l'activité d'un évènement malware.

— **Network-Based IPS** : Appareil dédié qui a la capacité de modifier ou jeter des paquets suspects, coupant les connexions TCP existantes.

Chapitre 11

Sécurité des logiciels

1 Problématique

1.1 Différents termes

— **CWE** : Common Weekness Enumeration (= erreur en général)
— **CVE** : Common Vulnerabilities and Exposures (= cas particulier dans un soft précis) => Maintenus par MITRE (= organisme de recherche américain soutenu, entre autre par le DoD)
— **SANS** : Centre international de formation à la sécurité informatique
— **OWASP** (Open Web Application Security Project) top ten (web applications security)
— **Software quality** : Contrairement à un matériel, un logiciel est un produit qui n'a pas une fiabilité prédictible, de plus il ne s'use pas dans le temps. Donc une anomalie survient ou ne survient pas dans l'exécution du logiciel, l'anomalie est présente de manière latente et peut ne jamais survenir (cfr Wikipédia). Le **Soft quality** est donc lié aux pannes accidentelles d'un programme résultant d'un input aléatoire, non anticipé. Afin de minimiser la fréquence d'occurrence des erreurs, on va tester un maximum d'input différents.
— **Programmation sécurisée** : c'est le fait de concevoir et d'implémenter un logiciel de telle sorte qu'il fonctionne même quand il est attaqué. Ce terme est lié à la **programmation défensive** qui est un état d'esprit qui consiste à écrire son code de façon à s'attendre au pire. Le fait est que le programmeur peut insérer des fautes non détectées ou des inconsistances. Pour s'en prémunir, il faut prévoir un traitement pour les fautes : soit en ajoutant du code vérifiant l'état du système, soit par un traitement d'erreur classique. Dans l'idéal, il faudrait penser à toutes les sources d'erreurs possibles et prévoir un traitement pour chacune d'elles (cfr journaldunet).

1.2 Interaction non sécurisées entre composants

1. Neutralisation incorrecte d'éléments spéciaux utilisés dans les commandes SQL (**Injection SQL**)
2. idem au niveau OS
3. idem au niveau des pages web avec le XSS (**Cross-site Scripting**)
4. la non-restriction d'upload de fichier à contenu dangereux

1.3 Mauvaise gestion des ressources systèmes

1. Télécharger un fichier sans vérification de son intégrité
2. Mauvais calcul de la taille d'un buffer
3. Format string non-contrôllé
4. Buffer Overflow

1.4 Mauvaise défense

1. Manque de cryptage pour les données sensibles/importantes
2. La non-restriction de tentatives excessives d'authentifications (blocage après 5 tentatives par exemple)
3. Exécution avec des privilèges non-nécessaire (non-application du principe de *Least Privilege*)

1.5 Les risques de sécurité critiques au niveau web

1. Injection
2. XSS
3. Mauvaise configuration de la sécurité
4. Redirection et forwards invalides
5. Exposition de données sensibles

2 Gestion des inputs

2.1 Inputs

Il s'agit de données venant de l'extérieur du programme, non connues du programmeur au moment de l'écriture du code. Ces derniers possèdent des caractéristiques (taille, type, contenu, ...) et viennent de diverses sources comme :

— Clavier
— Fichiers
— Réseau
— Environnement d'exécution
— OS

2.2 Buffer overflows

Définition

Une situation se produisant au niveau d'une interface, au niveau de laquelle il est possible de placer dans un **buffer** une quantité de données **dépassant la capacité allouée**, écrasant donc de l'information. Les attaquants exploitent cette situation pour crasher un système ou y insérer du code leur permettant de prendre le contrôle de la machine.

Exemple

Cfr slide 15 et 16.

Types de buffer overflows

1. **Stack buffer overflow** consiste à écraser les adresses de retour dans les appels de fonction
2. **Shellcode** est une chaîne de caractères qui représente un code binaire exécutable. À l'origine destiné à lancer un shell ('/bin/sh' sous Unix ou command.com sous DOS et Microsoft Windows par exemple), le mot a évolué pour désigner tout code malveillant qui détourne un programme de son exécution normale. Un shellcode peut être utilisé par un hacker voulant avoir accès à la ligne de commande par le transfert de l'exécution à du code fourni par l'attaquant (cfr Wikipedia).
3. **Heap overflow** est un bug semblable à un dépassement de tampon, mais contrairement à ce dernier où le débordement s'effectue dans la pile d'exécution du programme, ici le dépassement se fait dans le tas (la mémoire allouée dynamiquement lors de l'exécution d'un programme). Cfr wikipedia et vive la français de France bien français.
4. Global Data Area Overflow
5. Return to System Call

Défense contre les buffer overflows

1. **Compilation-time defense**. Il faut prévoir du code robuste face aux buffer overflows en fonction du choix du langage et en appliquant des techniques de codages sûres comme :
   — La programmation défensive
   — Eviter les librairies problématiques (gets, strings)
   — Manipuler les pointeurs et les buffers avec prudence et rigueur. Essayer de tous les cas de figure au niveau des inputs

2. **Runtime defense**. Permet de protéger du code existant, ce dernier potentiellement vulnérable. Pour se faire, on peut mettre en place certaines pratiques :
   — Ajustement des permissions sur l'espace d'adressage comme par exemple retirer l'exécution sur la stack (pile) ou le heap (tas)
   — Randomisation de l'espace d'adressage qui rendra plus difficile de deviner l'emplacement des morceaux de code vulnérables
   — Pages de garde qui consiste en des trous entre les différents composants de l'espace d'adressage

2.3 Interprétation de l'input

Problématique

Etant donné un input recu, comment l'interpréter et lui donner une signification ? Que ce dernier soit :
— **Textuel** ? à savoir un nom de fichier, URL, email, identifiant ?
— **Binaire** à savoir un entier, un flottant, une structure de données ?
Ils existent donc des risques comme les attaques par injection ou XSS notamment.

Attaque par injection

Les données entrées dans un programme peuvent influencer de manière accidentelle ou délibérée le flux d'exécution du programme. Il y a plusieurs types d'injection :
— injection de commandes (exemple slide 22-24)
— injection SQL (exemple slide 25)
— injection de code (exemple slide 26)

Attaque XSS

Afin de pouvoir gérer certaines applications Web, les navigateurs autorisent les scripts à accéder aux données d'autres pages que la page affichée à condition qu'il s'agisse du même site. Ceci crée une porte d'entrée pour les attaques XSS par le biais d'un accès au contenu de certains sites, à des cookies ou encore de Guest books, commentaires, forums.
A noter que la cible de ces attaques sont, non pas le site lui-même, mais les données qu'il contient/stocke (exemple de XSS slide 28-29).

Au niveau de la défense contre ces attaques, on pense à :
— la validation des inputs
— **ET** des ouputs

2.4 Validation de la syntaxe

— Vérification de la correspondance entre les données reçues et les données attendues. Pour cela, on peut utiliser les **expressions régulières** aka regex qui sont ultra puissantes (moins que One Punch Man quand même).
— Différencier les character sets (ASCII, unicode, ...). **Canonicaliser** (transformation dans une représentation minimale) pour permettre la validation.
— Utilisation de librairies de validation

2.5 Input fuzzing

Il s'agit d'une technique de test logiciel qui consiste à générer des inputs de test de manière aléatoire sans supposition initiale sur le type d'input attendu. Le but, ici, est de vérifier comment le programme va réagir avec des données anormales.
Il existe des outils en ligne qui sont utilisé également par les attaquants.

3 Ecrire du code sûr

Voici différents points à prendre en compte pour faire du code sécurisé :

1. Implémentation correcte de l'algorithme. Vérifier si les inputs sont légitimes est une chose mais il faut aussi s'assurer que le programme derrière gère correctement les inputs valides. Exemples à ne pas faire :
   — Utilisation d'un mauvais générateur de nombres aléatoires par Netscape pour générer les clés de sessions sécurisées
   — Génération prédictible des numéros de séquences TCP initiaux
   — Morris Worm, fin 80 : instructions de debug dans le programme sendmail utilisées pour contrôler le programme à distance
2. S'assurer que le langage machine généré corresponde à l'algorithme. Attention aux attaques sur les compilateurs.
3. Interpréter correctement les valeurs à l'aide de langage fortement typés. Faire attention aux casts (entier => pointeur en C).
4. Utilisation correcte de la mémoire. Attention aux fuites de mémoires qui entrainent des déni de service.
5. Attention à la programmation multithread. Race condition sur les ressources partagées => **deadlocks**.

3.1 Quelques termes

— A **lock** : occurs when multiple processes try to access the same resource at the same time.
— A **deadlock** occurs when the waiting process is still holding on to another resource that the first needs before it can finish.

Exemple :

Resource A and resource B are used by process X and process Y

1. X starts to use A.
2. X and Y try to start using B
3. Y 'wins' and gets B first
4. now Y needs to use A
5. A is locked by X, which is waiting for Y

The best way to avoid deadlocks is to avoid having processes cross over in this way. Reduce the need to lock anything as much as you can.

Sorry pour cette section mais j'avais vraiment pas envie de traduire et puis à l'Ephec on est tous bilingue (surtout Rémy) donc alaise.

4 Interaction avec l'OS

4.1 Least privilege

Principe à appliquer afin d'éviter l'escalade de privilèges. Pour se faire, il est idéal de ne donner des privilèges qu'aux portions de programmes qui en ont besoin (par exemple les serveurs qui ont besoin d'être root pour faire le bind sur un port privilégié).
Un autre exemple au niveau du serveur web, les droits (user www) doivent être limités en lecture, aux pages qu'il envoie, et en exécution, aux scripts qu'il est prévu d'exécuter. En effet, certains admins peu attentifs se contentent parfois de donner les droits sur le répertoire des pages web et à toute la hiérarchie qui en découle.

4.2 Autres points

1. Utilisation des appels systèmes et des librairies standards. Il faut faire l'effort de comprendre les fonctions qu'on utilise afin d'être certain qu'elles fassent vraiment ce qu'on veut qu'elles fassent.
2. Interaction avec d'autres programmes. Il faut, ici, se renseigner sur les aspects programmations (programmation défensive ? Oui ou non), sur les aspects de sécurité et enfin sur les flux de données utilisées par le programme si il en utilise afin de s'assurer d'utiliser un programme digne de nous, digne des Enfants Terribles, digne de Snake quoi.

Chapitre 12

Sécurité des OS

1 Planning

1.1 Top 4 des stratégie de minimisation de risques

L'implémentation de ce top 4 (de l'**ASD** Top 35 Mitigation Strategies) aurait suffit à prévenir au moins 85% des instrusions depuis 2010 :

1. White-list d'applications approuvées
2. Patch des vulnérabilités des OS et des applications tierces
3. Restriction des privilèges administrateur
4. Création d'une défense système en profondeur

1.2 Processus de déploiement d'un système

Les systèmes étant vulnérables dès le processus d'installation, il est important de planifier une certaine procédure :

1. Evaluer les risques et planifier le déploiement
2. Sécuriser l'OS et les applications clés
3. S'assurer que le contenu critique est sécurisé
4. S'assurer que les mécanismes de protection réseau sont utilisés
5. S'assurer que les processus appropriés sont utilisés pour maintenir la sécurité

1.3 Quelques considérations

Avant de procéder au hardening de sa machine, il faut se poser quelques questions afin d'avoir en tête ce qui est à protéger et la façon de protéger. Ces questions sont entre autre :

- Quel est le but du système ? Quelles infos stocke-t-il ? Fournit-il des services ? Quel est le niveau de sécurité nécessaire ?
- Déterminer les catégories d'utilisateurs ainsi que les privilèges.
- Quid de l'authentification des utilisateurs ?
- Quid de l'administration du système ?
- Comment gerer l'accès à l'informations présente sur le système ? (BDD, serveurs de fichiers, ...)
- Faut-il des mesures de sécurité supplémentaires tel qu'un firewall, un anti-virus, anti-malwares, ... ?

2 Hardening

Pour mettre en place la sécurisation d'un système, plusieurs étapes sont à envisager :

1. Lors de l'installation, effectuer la configuration et le patching à savoir :
   - Accès réseau initial. De préférence, ce dernier doit être minimal, voire non existant. Le minimum correspond à l'accès vers l'extérieur, vers des sites web bien spécifiques
   - installer uniquement les packages nécessaires

— Faire attention aux drivers
— Sécurisation du démarrage à savoir les options du BIOS, mot de passe, les médias de boot, ...
— Patchs de sécurité en faisant attention aux updates automatiques

2. Nettoyage des services, applications et protocoles. Ici, il faut faire attention à éviter les installations par défaut, d'installer à posteriori plutôt qu'à priori (mieux vaut ne pas installer plutôt que désinstaller ou désactiver)
3. Configurer les utilisateurs, groupes et l'authentification en tenant compte du principe de *Least Privilege*, que les droits d'admin se résument aux tâches qui le nécessitent vraiment, de désactiver des comptes par défaut et de la gestion des mot de passes.
4. Configurer le contrôle des ressources à savoir la configurations des permissions sur les ressources : exécution de programmes ou l'accès aux fichiers
5. Installation de contrôle de sécurité supplémentaires comme par exemple un anti-virus ou une white-list d'applications.
6. Enfin il faut tester la sécurité du système. A savoir :
   — Vérifier l'implémentation correction des mesures prises précédemment
   — Identifier les vulnérabilités restantes à corriger
   — Check-list (pas des pings comme Sergen svp) disponibles dans les guides de hardening
   — Outils de scan de vulnérabilités comme *nessus*

   Tout ceci était à faire après le hardening initial et à reproduire de manière régulière si possible.

3 Sécurité applicative

Après avoir effectué l'installation du système, il faut envisager l'ajout des services et applications nécessaire en faisant attention aux services fournissant un accès distant, aux scripts d'installation par défaut et aux comptes créées de manière automatique.

Slide de merde..

4 Maintenance

1. Mettre en place une politique de patching et d'update
2. Evaluer régulièrement les vulnérabilités potentielles
3. Disk logging : Gestion de l'espace disque et analyse
4. Backup et archivage des données. Détermination lors de la planification dans l'idéal et déterminer la localisation du backup (local, distant, je sais pas moi, ...)

Voici un mauvais exemple de politique de backup :
— Hébergeur australien attaqué en 2011, les attaquants ont détruit l'ensemble des sites hébergés mais également les copies de backup stockées en ligne
— Backups gardés sur le site : Perdus en cas d'inondation ou d'incendie du centre ID.

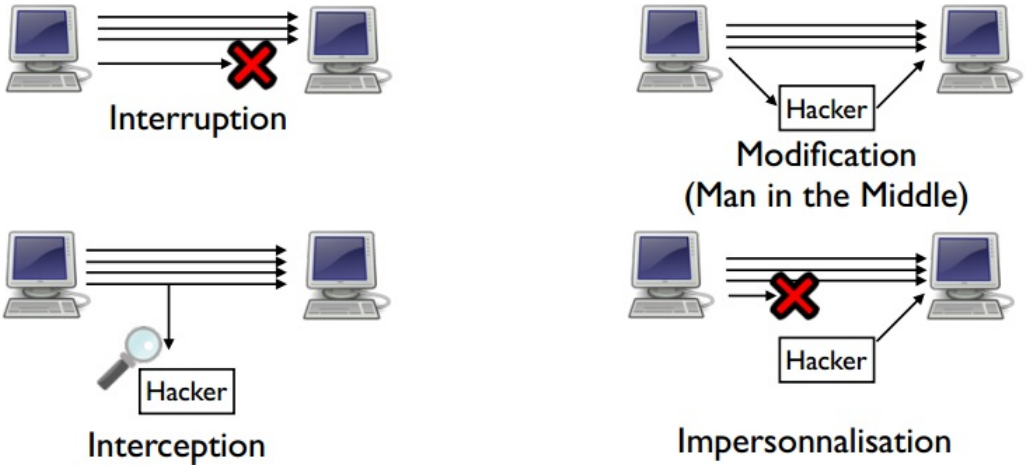
Chapitre 13

Protocoles et standards de sécurité Internet

1 Couche réseau

Le but initial du protocole IP est d'assurer la disponibilité de la communication mais quand est-il de la confidentialité et de l'intégrité des données ?

1.1 Types d'attaques



1.2 Attaques ICMP

- *Host Unreachable* qui provoque la déconnexion des sessions de la victime
- *ICMP Redirect* qui permet de dévier les paquets de la victime
- **Ping flood** est une forme simple d'attaque par déni de service, où l'attaquant inonde le serveur cible de requêtes ping. Ce type d'attaque ne réussit que si l'attaquant a plus de bande passante que sa victime (par exemple, un hacker avec une connexion Internet qui transmet 20 millions de bits par seconde et une victime avec une connexion Internet de 10 millions de bits par seconde). Cfr wikipedia
- **Ping of Death** aka Le ping de la mort est une attaque historique de type déni de service réalisé par l'envoi de paquet ping malformé. Un ping a normalement une taille de 56 octets (soit 84 octets avec l'entête IP), or certains systèmes n'étaient pas en mesure de traiter correctement les paquets plus gros que la taille maximale (65 535 octets) pouvant provoquer un crash de la machine cible. Cfr wikipedia
- **Smurf** consiste à ping sur une adresse de broadcast avec une IP source spoofée. Qui plus est, les attaques par rebond constituent une famille d'attaques de système d'information qui consistent à utiliser un ou des systèmes intermédiaires, participant à leur insu, et permettant à un assaillant de rester caché. cfr Wikipedia

1.3 ARP Poisoning/Spoofing

L'**ARP spoofing** ou **ARP poisoning** est une technique utilisée en informatique pour attaquer tout réseau local utilisant le protocole de résolution d'adresse ARP, les cas les plus répandus étant les réseaux Ethernet et Wi-Fi. Cette technique permet à l'attaquant de détourner des flux de communications transitant entre une machine cible et une passerelle : routeur, box, etc. L'attaquant peut ensuite écouter, modifier ou encore bloquer les paquets réseaux. cfr Wikipedia notre ami

- Interception, MITM ou DoS
- Insérer des fausses informations dans les tables APR pour dévier le trafic
- **Gratuitous ARP** : l'attaquant émet une trame ARP en broadcast (à tout le réseau) dans laquelle il fait correspondre son adresse MAC à l'adresse IP de la passerelle. Le gratuitous ARP est initialement prévu pour que les équipements venant d'arriver sur le réseau s'annoncent (ce qui permet par exemple de détecter les IP dupliquées). Ce type de requête très utilisé par des équipements de réseau n'est pas mauvais en soi mais pourrait être détourné si les destinataires sont très mal protégés.
- **ARP forgée** : l'attaquant émet une requête en unicast vers la victime en spécifiant comme adresse IP émettrice, l'adresse IP qu'il veut usurper et en indiquant sa propre adresse MAC comme l'adresse MAC de l'émetteur. Ainsi, lorsque la victime reçoit la requête, elle enregistre la correspondance IP/MAC dans sa table ARP alors que celle-ci est erronée.

1.4 Attaques sur IP

Une **attaque par fragmentation** (en anglais fragment attack) est une attaque réseau par saturation (déni de service) exploitant le principe de fragmentation du protocole IP.

En effet, le protocole IP est prévu pour fragmenter les paquets de taille importante en plusieurs paquets IP possédant chacun un numéro de séquence et un numéro d'identification commun. A réception des données, le destinataire réassemble les paquets grâce aux valeurs de décalage (en anglais offset) qu'ils contiennent. cfr commentcamarche

- **IP Fragment Overlap** : The IP fragment overlapped exploit occurs when two fragments contained within the same IP datagram have offsets that indicate that they overlap each other in positioning within the datagram. This could mean that either fragment A is being completely overwritten by fragment B, or that fragment A is partially being overwritten by fragment B
- **IP Fragmentation Buffer Full** : The IP fragmentation buffer full exploit occurs when there is an excessive amount of incomplete fragmented traffic detected on the protected network.
- **IP Fragment Overrun** : The IP Fragment Overrun exploit is when a reassembled fragmented datagram exceeds the declared IP data length or the maximum datagram length.
- **IP Fragment Too Many Datagrams** : The Too Many Datagrams exploit is identified by an excessive number of incomplete fragmented datagrams detected on the network.
- **IP Fragment Incomplete Datagram** : This exploit occurs when a datagram can not be fully reassembled due to missing data. This can indicate a denial of service attack or an attempt to defeat packet filter security policies.
- **IP Fragment Too Small** : If an IP fragment is too small it indicates that the fragment is likely intentionally crafted. Any fragment other than the final fragment that is less than 400 bytes could be considered too small.

1.5 Contre-mesures

1. Au niveau d'un firewall, filtrer les ICMP
2. Valider les adresses sources en sortie du réseau
3. Limiter l'usage de broadcast
4. Au niveau ARP : un table ARP statique, IDS, blocage des ARP gratuits, surveillance
5. IP : Chiffrement avec IPSec

1.6 IPSec

Définition

IPSec est un ensemble de protocoles permettant le transport de données sécurisées sur un réseau IP. Il permet notamment l'authentification et le chiffrement des paquets IP.
A la base, ce dernier a été développé pour IPv6, puis adapté à IPv4. Cette sécurisation est également

pour tout ce qui est liaison VPN.

Mode d'utilisations

Il existe 2 modes d'utilisations possibles :

1. **Mode tunnel** : la totalité du paquet IP est chiffré et authentifié, puis encapsulé dans un nouveau paquet IP avec un nouvelle en-tête. Ce mode est utilisé pour le VPN et traverse les NATs.
2. **Mode transport** : uniquement le payload est chiffré et authentifié. Le routage n'est donc pas impacté ici. Ce mode est utilisé pour des communications hôte à hôte mais ne traverse pas les NATs avec AH.

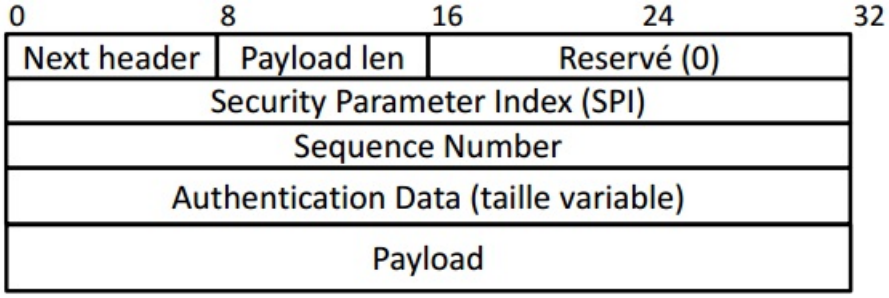
Initialisation de la connexion

IPSec initie ses connexions de manière logique => **Security Association**.

1. Authentification et échange de clés par le protocole IKE (Internet Key Exchange)
   — Authentification sur base d'un secret partagé
   — ou sur base de crypto asymétrique
2. Permet l'établissement d'une Security Association qui est défini par :
   — l'identifiant du protocole de sécurité (AH ou ESP)
   — L'adresse IP source
   — Un identifiant de 32 appelé SPI (Security Parameter ID), qui sera réutilisé dans tous les paquets de l'échange

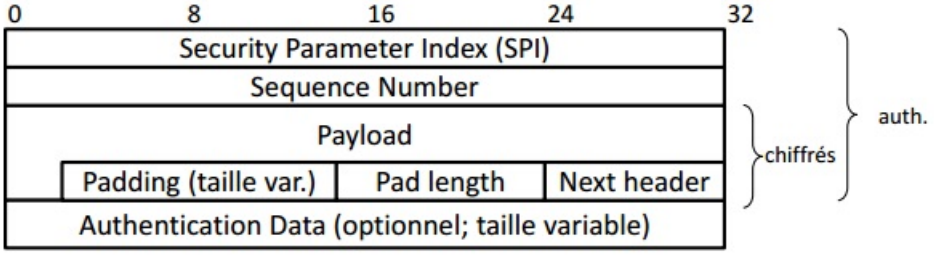
Authentification Header Protocol (deprecated)

Ce protocol garantit l'authentification de la source et l'intégrité des données, mais pas la confidentialité.
Après l'établissement de la Security Association, un en-tête AH est inséré entre le payload et l'en-tête du paquet.



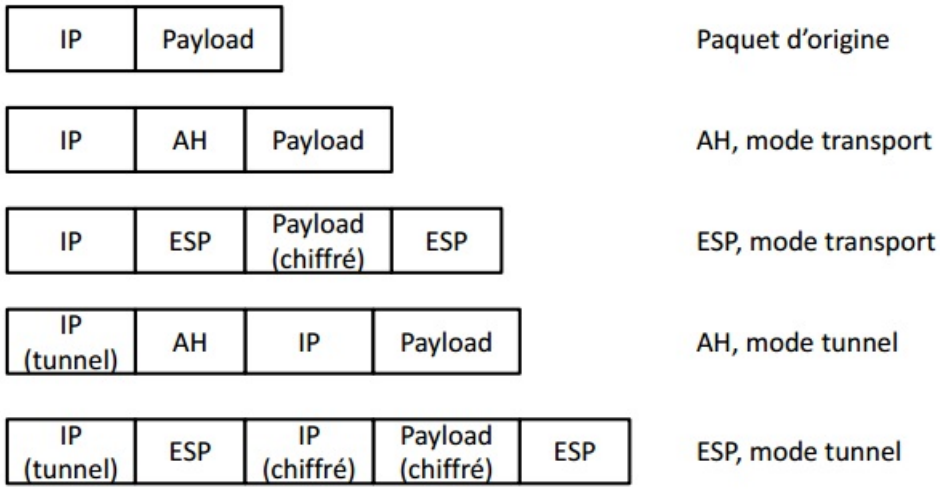
Encapsulating Security Protocol

Comme AH, l'en-tête ESP s'insère entre l'en-tête et le payload IP. Ce dernier possède en plus un trailer. A savoir que certains champs de l'en-tête sont chiffrés.



Résumé

Il parait que si on comprend ce schéma, on a tout compris.



2 Couche Transport

2.1 Les attaques possibles

1. **UDP Flood** : Ce déni de service exploite le mode non connecté du protocole UDP. Il crée un "UDP Packet Storm" (génération d'une grande quantité de paquets UDP) soit à destination d'une machine soit entre deux machines. Une telle attaque entre deux machines entraîne une congestion du réseau ainsi qu'une saturation des ressources des deux hôtes victimes. La congestion est plus importante du fait que le trafic UDP est prioritaire sur le trafic TCP. En effet, le protocole TCP possède un mécanisme de contrôle de congestion, dans le cas où l'acquittement d'un paquet arrive après un long délai, ce mécanisme adapte la fréquence d'émission des paquets TCP et le débit diminue. Le protocole UDP ne possède pas ce mécanisme. Au bout d'un certain temps, le trafic UDP occupe donc toute la bande passante, ne laissant qu'une infime partie au trafic TCP. cfr Wikipedia
2. **Land Attack** : attaque réseau datant de 1997, utilisant l'usurpation d'adresse IP afin d'exploiter une faille de certaines implémentation du protocole TCP/IP dans les systèmes. Le nom de cette attaque provient du nom donné au premier code source (appelé « exploit ») diffusé permettant de mettre en oeuvre cette attaque : land.c. L'attaque LAND consiste ainsi à envoyer un paquet possédant la même adresse IP et le même numéro de port dans les champs source et destination des paquets IP. Dirigée contre des systèmes vulnérables, cette attaque avait pour conséquence de faire planter les systèmes ou de les conduire à des états instables.
3. **SYN Flooding**
4. **Session Flooding** : Etablissement d'un grand nombre de connexion TCP pour saturer la table
5. **TCP Sequence Guessing** : A TCP sequence prediction attack is an attempt to predict the sequence number used to identify the packets in a TCP connection, which can be used to counterfeit packets. The attacker hopes to correctly guess the sequence number to be used by the sending host. If they can do this, they will be able to send counterfeit packets to the receiving host which will seem to originate from the sending host, even though the counterfeit packets may in fact originate from some third host controlled by the attacker
6. **Scanning de port** : Découvrir l'état (open, close, drop) des services d'un hôte, et/ou OS fingerprinting (procédé permettant de déterminer l'identité du système d'exploitation utilisé sur une machine distante en analysant les paquets provenant de cet hôte)
7. **SYN scan :** on déduit que si un ACK est bien reçu par un service que ce dernier est bien en écoute
8. **ACK scan :** si on reçoit un RST alors on est dans un état open sinon c'est un état filtré
9. FIN Scan, UDP Scan, Scan quoi, ...

2.2 Contre-mesures

— contre le SYN Flood, utilisation d'IPSec ou de SYN Cookies (valeurs particulières des numéros de séquences initiales générés par un serveur (ISN : Initial Sequence Number) lors d'une demande de connexion TCP)
— contre les scans, utilisation de firewall ou bien d'IDS
— Sécurisation des connexion TCP par la mise en place de SSL/TLS

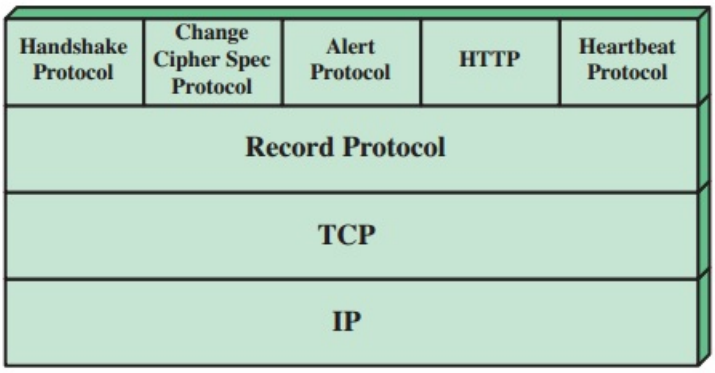
2.3 Transport Layer Security

Définition

Protocole de sécurisation des échanges sur Internet fonctionnant suivant un mode client-serveur, il a pour objectif de fournir un service sécurisé de bout-en-bout avec TCP.

Les objectifs de sécurité sont les suivants :
— l'authentification du serveur
— la confidentialité des données échangées (ou session chiffrée)
— l'intégrité des données échangées
— de manière optionnelle, l'authentification du client (mais dans la réalité celle-ci est souvent assurée par le serveur)

Architecture



Connexion TLS

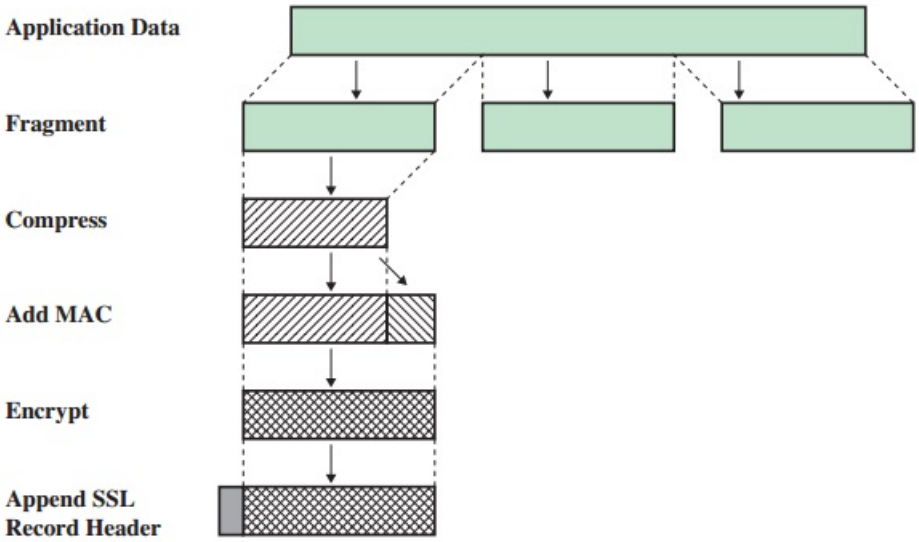
Relation transitoire au niveau de la couche transport qui fournit un type de service. Une connexion est associée à une session.

Session TLS

Association entre un client et un serveur. Définit un ensemble de paramètres de sécurité cryptographique qui peuvent être partagés par plusieurs connexions.

Record Protocol

Permet de fournir la confidentialité et l'intégrité des messages.



Change Cypher Spec Protocol

- The change cipher spec message is sent by both the client and server to notify the receiving party that subsequent records will be protected under the just-negotiated CipherSpec and keys.
  - It exists to update the cipher suite to be used in the connection.
  - It permits a change in the SSL session occur without having to renegotiate the connection.
  - The message consists of a single byte of value 1.
- There are two states for the change cipher spec message.
  - Read Current
  - Read Pending
- The change cipher spec message is normally sent at the end of the SSL handshake.

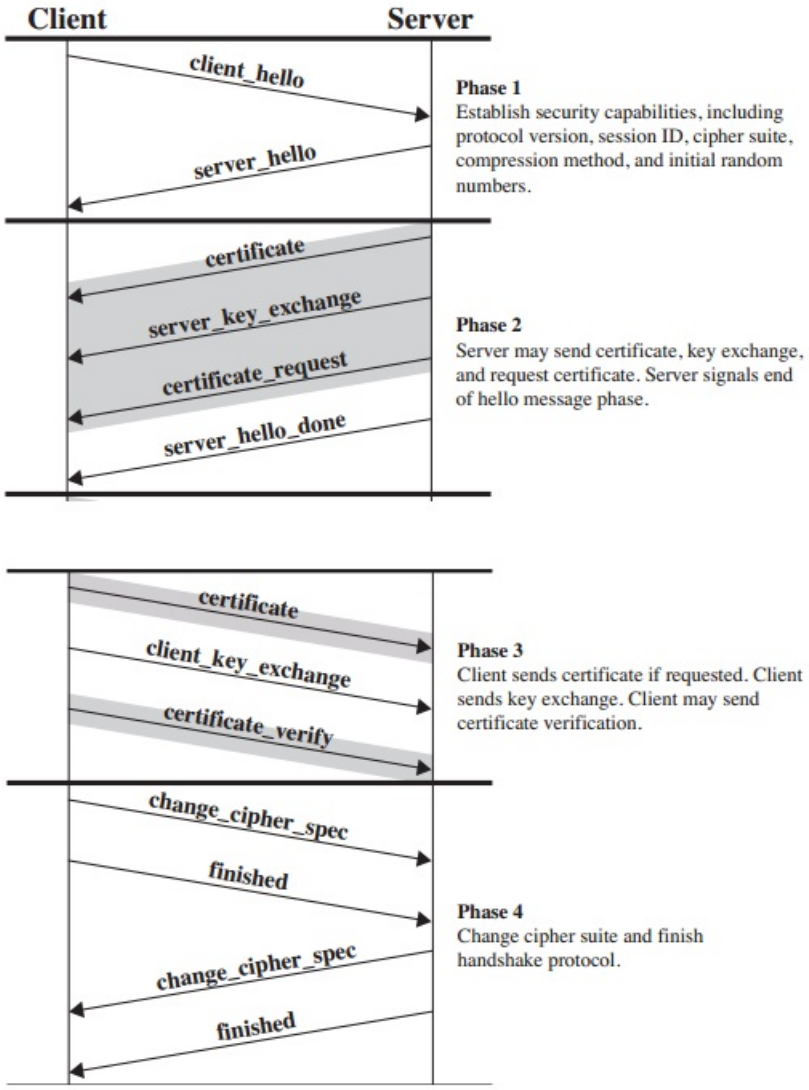
Alert Protocol

Permet l'envoi de notification TLS au pair. Ce dernier contient 2 octets :
- Le premier prend la valeur 1 (warning) ou 2 (fatal) pour indiquer la sévérité du message
- Le second contient un code indiquant de quelle alerte il s'agit (ex : MAC incorrect, message closenotify)

Handshake Protocol

Permet au client et au serveur de s'authentifier, de négocier les algorithmes MAC et de chiffrement, ainsi que les clés de chiffrement.
Le *handshake* se déroule en 4 phases :

1. Initier la connexion logique et établir les capabilities de sécurité
2. Le serveur peut envoyer un certificat, initier un échange de clé et demander un certifiant au client
3. Le client vérifie le certificat du serveur et envoie ses propres informations
4. Envoi des messages changecypherspec, et début de l'échange applicatif



Heartbeat Protocol

Permet de surveiller la disponibilité des entités du protocole. Il possède 2 types de messages :
— heartbeat$ _ $request
— heartbeat$ _ $response
Enfin ce protocole a pour objectif de s'assurer que le receveur est toujours actif et de générer de l'activité pendant des périodes d'inactivité pour éviter les timeouts dans les firewalls.

3 Couche applicative

3.1 HTTPS

Il s'agit simplement de la combinaison de **HTTP** et de **SSL** afin de créer un canal de communication sécurisé entre un navigateur web et un serveur web.
Il existe 3 niveaux de connexions, à savoir :
— Requête de connexion HTTP
— Etablissement de session (et d'une ou plusieurs connexion) SSL/TLS
— Etablissement de la connexion TCP
Lors de la fermeture d'une connexion de ce genre, il faut bien faire attention à bien gérer la fin de *toutes* les connexion.

3.2 Mail

Quelques menaces

1. **Confidentialité**. Les protocoles POP, IMAP et SMTP en clair
2. **Intégrité**. Usurpation d'identité à l'envoi ou durant le transit
3. **Spam**
4. **Phishing**

Le spam

Le spam, courriel indésirable ou pourriel (terme recommandé au Québec par l'OQLF1) est une communication électronique non sollicitée, en premier lieu via le courrier électronique. Il s'agit en général d'envois en grande quantité effectués à des fins publicitaires.

Cette pratique exerce un impact. Notamment au niveau de :
— **Infrastructure**. Bande passante et mémoire au niveau des serveurs et des mailboxes
— **Productivité**. En effet, le spam engendre une perte de temps (tri des messages), une perte d'argent (cout du spam jusqu'à 1000$/an/employé).

On a estimé, en 2008, que 92% des emails étaient du spam. Pour en finir avec les chiffres, le coût global estimé pour lutter contre cette plaie s'élève à 100 milliards de dollards/an.

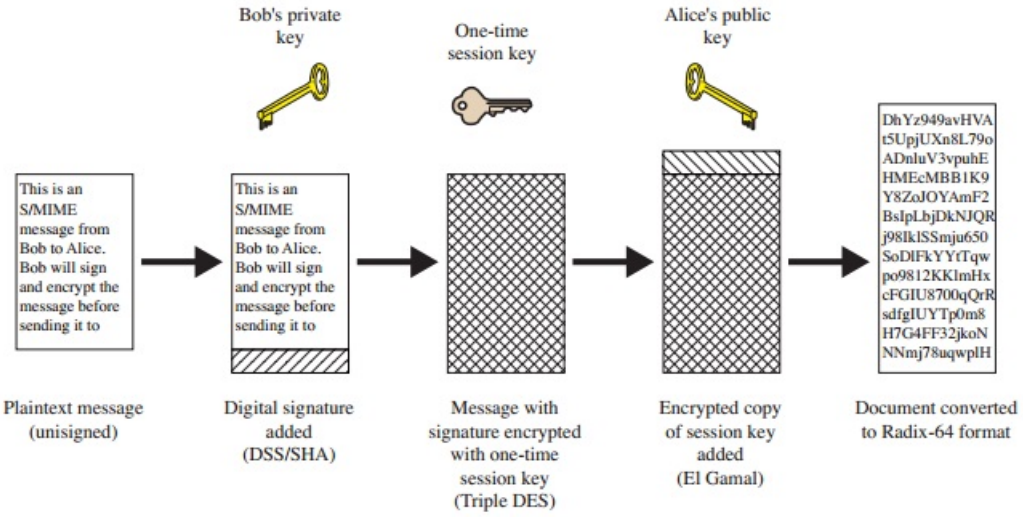
Il existe, évidemment, différentes techniques afin de spammer pour casser les couilles des pauvres secrétaires. On cite entre autre :
— **Spoofing** d'adresses sources
— **Open relays**. En français, relais non protégé. Il s'agit d'un serveur de messagerie qui n'a pas été correctement configuré et qui permet alors aux spammeurs de l'utiliser, à l'insu de l'administrateur du serveur, pour effectuer leurs envois en masse. Les spammeurs masquent ainsi leur identité et peuvent contourner des outils de filtrage puisque leurs messages sont émis depuis ce serveur intermédiaire qui lui, n'est pas blacklisté - mais le deviendra rapidement dès qu'il sera découvert.
— Ouverture automatique de comptes email
— **Botnets** pour bombarder de mails
— Collecte d'adresses de destination probables : crawlers, achats, attaques par dictionnaire, hacking, virus/spywares, hoax/chaine de lettres, ... Ici, le but est de vérifier qu'une adresse mail est valide et active. Le connard qui répond à un hoax ou à une chaine de lettres sera listés d'office dans la base de données des spammers. Ce connard n'aura plus qu'à changer d'adresses mail.

Depuis le temps, des contre-mesures ont bien sûr été trouvées et mises en place pour lutter contre les spammers.
On pense notamment à :
— La mise en place de filtre basés sur le format et sur le contenu du message.
— White/Black list
— Au niveau législatif. Plusieurs condamnations de parrains du spam car, Oui, le spam est illégal.
— les protocoles DKIM, PGP et S/MIME.

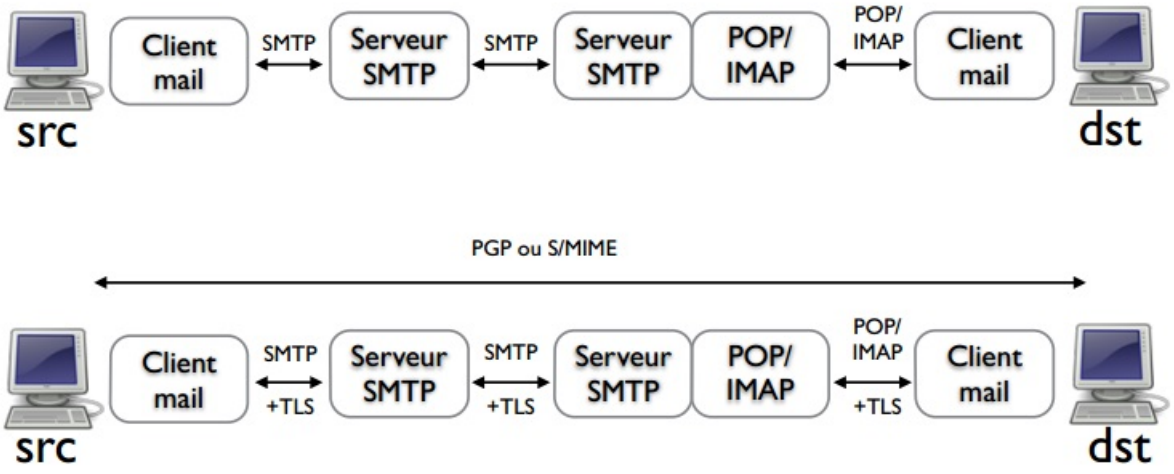
Sécurisation

Il existe un protocole qui permet la signature cryptographique de message, par domaine. Ce dernier apporte une protection plutôt efficace contre le spam et le phishing en assurant l'intégrité, l'authentification, la non-répudiation et la confidentialité des données. Il s'agit du protocole **DKIM** qui est une norme d'authentification fiable du nom de domaine de l'expéditeur d'un courrier électronique.

Un autre protocole est **S/MIME** qui fournit la possibilité de signer et/ou de chiffrer des messages emails par l'utilisation de la cryptographie asymétrique.
Il possède 4 fonctions :
— Données sous enveloppe : **Chiffrement**
— Données signées : **Signature numérique**, message encodé en base64
— Données signées en clair : **Signature numérique**, message non encodé
— Donnée signées et sous enveloppe



Enfin, un dernier protocol de sécurisation des emails est **Pretty Good Privacy**. C'est un logiciel de cryptographie permettant de garantir la confidentialité et l'authentification pour les communications des données. Ce dernier est entre autre utilisé pour la signature des données et le chiffrement des fichiers, partitions ou emails. Il fait parti des logiciels utilisant la cryptographie hybrides (symétrique et asymétrique)



3.3 DNS

Quelques attaques

1. **DNS Poisoning**. L'idée est de forcer un serveur DNS à mettre en cache des records malfaisants. Ce type d'attaque permet, par exemple, d'envoyer un utilisateur vers un faux site dont le contenu peut servir à de l'hameçonnage (dans le cas du DNS, on parle de pharming) ou comme vecteur de virus et autres applications malveillantes. A savoir qu'à la base, le DNS ne permettait pas de faire de vérification du lien entre la requête et la réponse ce qui facilitait l'attaque. Dorénavant, une vérification est effectuée. Cependant, il est toujours possible de *forcer* des utilisateurs à effectuer des requêtes par le biais de liens dans un mail par exemple.
2. **DoS**. L'idée est de forcer un serveur à envoyer des réponses DNS vers une machine envoyant des requêtes utilisant l'adresse IP de la cible comme source. Avec cette attaque, on constate un effet d'amplification. En effe,t les réponses DNS sont plus grandes que les requêtes (facteur d'amplification jusqu'à 60).

Sécurisation

Utilisation du protocole DNSSec permettant de résoudre certains problèmes de sécurité liés au protocole DNS. Ce dernier permet notamment de :

— Sécuriser les données envoyées par le DNS. Contrairement à d'autres protocoles comme SSL, il ne sécurise pas juste un canal de communication mais il protège les données, les enregistrements DNS, de bout en bout. Ainsi, il est efficace même lorsqu'un serveur intermédiaire trahit.
— signer cryptographiquement les enregistrements DNS et met cette signature dans le DNS. Ainsi, un client DNS méfiant peut récupérer la signature et, s'il possède la clé du serveur, vérifier que les données sont correctes. Cela permet de prévenir le cache poisoning

Il est intéressant à signaler que ce protocole est toujours en cours de déploiement donc n'est pas encore présent sur tout les serveurs DNS.

4 Authentification sur Internet

4.1 Problématique

Comment faire pour authentifier des hôtes à travers un réseau non sécurisé ?
— One-time password (digipass, smart cards, ...)
— Biométrique
— Logiciel d'authentification lié à un serveur d'authentification sécurisé

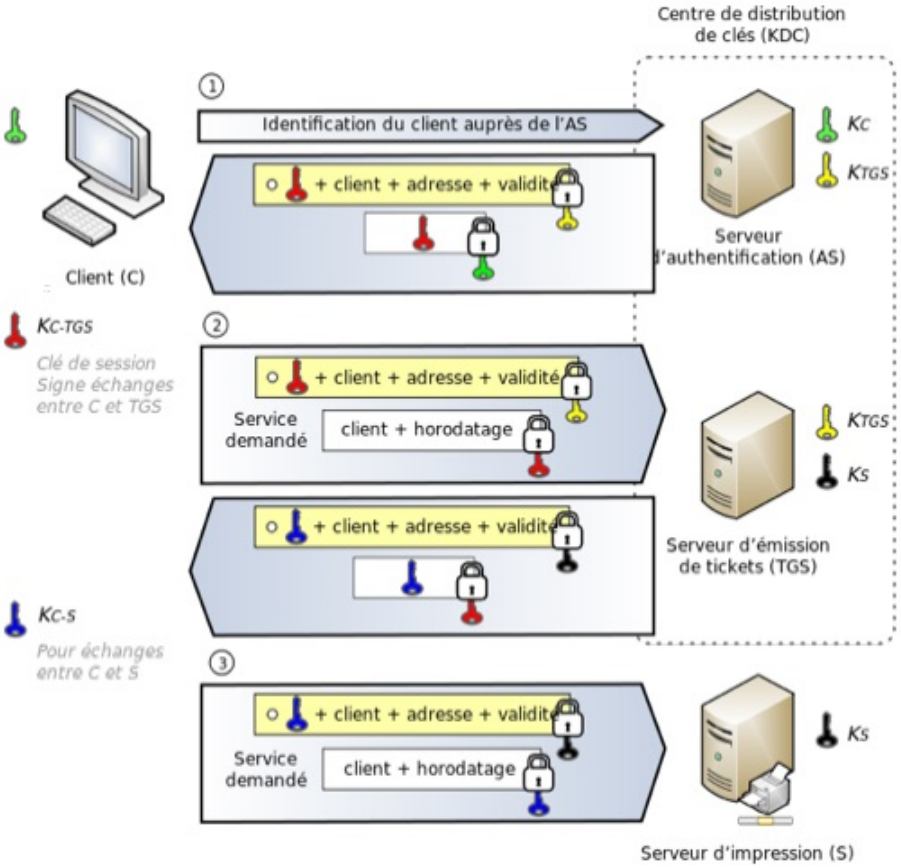
4.2 Kerberos

Définition

Protocole d'authentification réseau qui repose sur un mécanisme de clés secrètes (chiffrement symétrique) et l'utilisation de tickets, et non de mots de passe en clair, évitant ainsi le risque d'interception frauduleuse des mots de passe des utilisateurs.

Ce que permet ce protocole :
— Authentification de communications client/serveur
— Authentification mutuelle
— Protège contre l'écoute (eavesdropping) et contre les attaques par rejeu
— Utilisation d'un tiers de confiance (serveur d'authentification)
— Utilisation de tickets

Principes

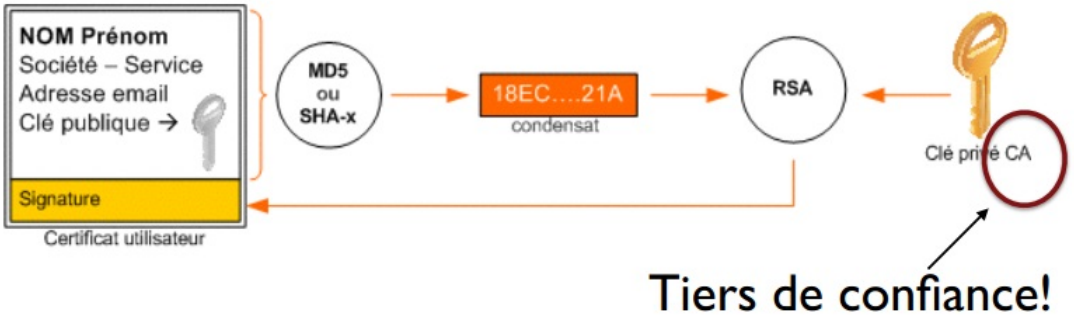


4.3 Public-Key Infrastructure

Définition

Une infrastructure à clés publiques ou Public Key Infrastructure (PKI), est un ensemble de composants physiques (des ordinateurs, des équipements cryptographiques logiciels ou matériel type HSM ou encore des cartes à puces), de procédures humaines (vérifications, validation) et de logiciels (système et application) utilisés pour créer, gérer, stocker, distribuer et révoquer des certificats numériques sur base de la cryptographie asymétrique.

Une PKI offre un ensemble de services :
— Enregistrement des utilisateurs ou des équipements
— Génération, publication, renouvellement ou révocation de certificats
— Publication de listes de révocation
— Identification et authentification des utilisateurs



Ici, les autorités de certification (CA) ont un rôle à jouer. En effet, après vérification de l'identité du demandeur, le CA signe, émet et maitient les certificats et les listes de révocations.
Ces derniers doivent également être authentifiés. Ils disposent de leurs propres certificats, qui doivent

eux aussi être signés par un CA (notion de **Hiérarchie**). Au sommet de cette hiérarchie, on retrouve les **Certificats Racine** qui sont inclus dans les navigateurs/OS.

4.4 OpenID

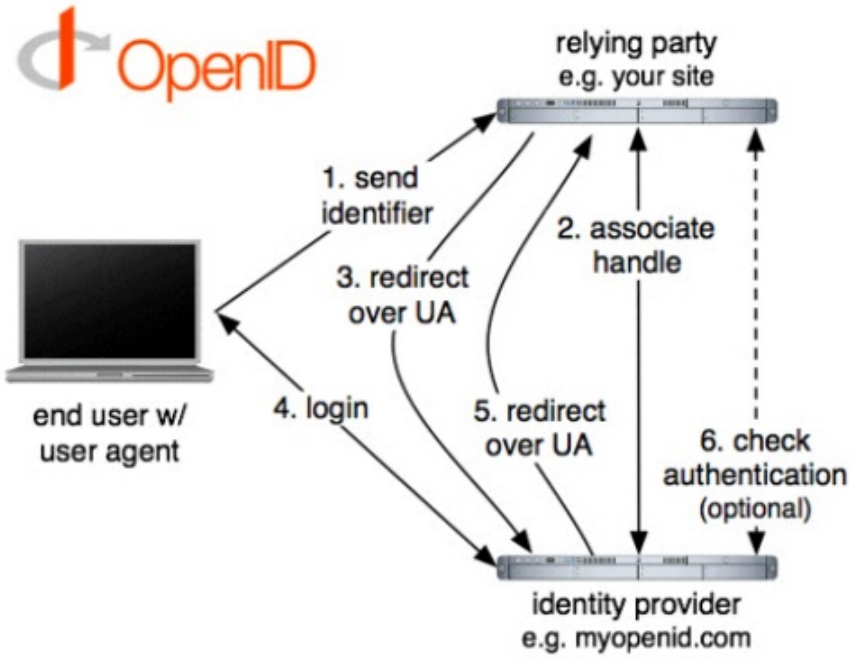
Définition

C'est un système d'authentification centralisée permettant l'authentification unique. Il permet à un utilisateur de s'authentifier auprès de plusieurs sites (devant prendre en charge cette technologie) sans avoir à retenir un identifiant pour chacun d'eux mais en utilisant à chaque fois un unique identifiant OpenID. Le modèle se base sur des liens de confiance préalablement établis entre les fournisseurs de services et les fournisseurs d'identité (OpenID providers). Il permet aussi d'éviter de remplir à chaque fois un nouveau formulaire en réutilisant les informations déjà disponibles.

Intérêts

1. Permet, pour un gestionnaire de site web, de déléguer l'authentification auprès d'un fournisseur d'identité, ou OpenID Provider
2. Le site Web et le fournisseur créent un secret partagé
3. L'utilisateur se loge auprès du fournisseur d'identité, et ce dernier lui fournit une preuve cryptographique de l'authentification
4. Le site web peut vérifier l'authentification grâce au secret partagé avec le fournisseur

Schéma



Chapitre 14

Gestion de la sécurité

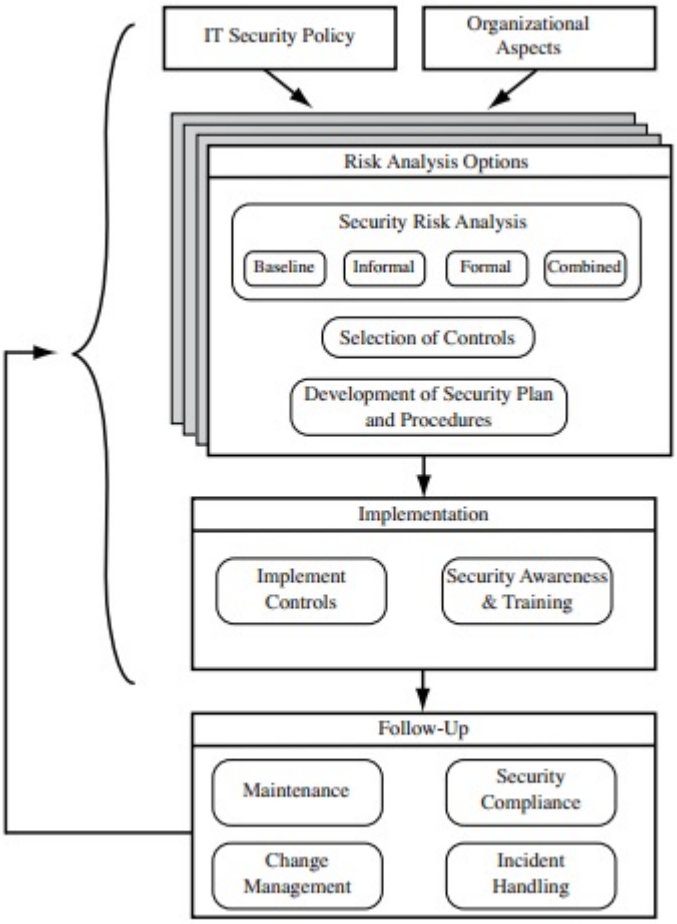
1 Définition

Par la **gestion de la sécurité**, on entend le processus utilisé pour réaliser et maintenir un certain niveau de confidentialité, d'intégrité, de disponibilité, d'authentification, de traçabilité et de fiabilité.

Il y a 3 questions fondamentales qu'il faut se poser :
— Quels sont les avoirs à protéger ?
— En quoi sont-ils menacés ?
— Que pouvons-nous faire pour contre ces menaces ?

Par gestion de la sécurité, on entend également :
— Déterminer les objectifs, les stratégies et les politiques de sécurité de l'organisation
— Déterminer les requirements de sécurité
— Identifier et analyser les menaces sur les avoirs
— Identifier et analyser les risques
— Spécifier des mesures de protection appropriées
— Surveiller l'implémentation et le fonctionnement de ces mesures
— Détecter et réagir aux incidents

Il existe différents **standards de sécurité** qui représentent un consensus sur les bonnes pratiques dans le domaine de la sécurité. On pense notamment aux standards ISO, NIST, ...



2 Procédure

1. **Analyse des risques**. Définir ce que le mécanisme doit faire, et ce qu'il doit protéger :
   (a) Valeur des avoirs
   (b) Vulnérabilités
   (c) Probabilité et impact =$>$ Calcul des risques
2. Mettre en place des **contre-mesures/contrôles** :
   (a) Prévention, Détection, Réaction, Récupération
   (b) Choix stratégique : Budget étant limité, il faut avoir recours à une sélection des contre-mesures sur base de l'analyse des risques
   (c) Ne pas oublier les risques résiduels
3. **Validation** : Comment s'assurer que la contre-mesure fonctionne ?

Voici un exemple d'analyse des risques :

Asset	Threat/ Vulnerability	Existing Controls	Likelihood	Consequence	Level of Risk	Risk Priority
Internet router	Outside hacker attack	Admin password only	Possible	Moderate	High	1
Destruction of data center	Accidental fire or flood	None (no disaster recovery plan)	Unlikely	Major	High	2

3 Contrôles

3.1 Définition

Il s'agit d'action, appareil, procédure ou tout autre mesure qui réduit le risque en éliminant ou en empêchant une violation de la sécurité, en minimisant le danger qu'elle peut causer ou en découvrant et signalant son apparition pour permettre une action corrective.

3.2 Types de contrôles

— **Contrôles de gestion** : Procédures, politiques et processus guidant la stratégie de sécurité
— **Contrôles opérationnels** : S'occupent de l'implémentation et de l'application des politiques et des standards de sécurité
— **Contrôles techniques**

Ces contrôles peuvent agir à différents niveaux/moments :
— **Contrôles de soutien** : *mesures de base* utilisées par les autres contrôles
— **Contrôles préventifs** : Empêche les tentatives de violation
— **Contrôles de détection et de recovery** : En réponse aux attaques

3.3 Implémentation des contrôles

Deux étapes, à savoir :

1. Implémentation du plan en lui-même ainsi que les validations :
   — Suivi des dépenses
   — Mise en place correcte des contrôles
   — Surveillance et administration des contrôles
2. Formation et conscientisation à la sécurité. => Le personnel impacté par les contrôles doit être formé à l'utiliser correctement

3.4 Validation et monitoring

Les contrôles doivent être constamment surveillés pour s'assurer de leur bon fonctionnement. Certaines pratiques résident là-dessous :
— Maintenance
— Adéquation par rapport au plan de sécurité
— Gestion des changements
— Gestion des configurations
— Réaction aux incidents

4 Plan de sécurité IT

Après avoir identifié les contrôles susceptibles de diminuer un maximum les risques de sécurité identifiés, un **plan de sécurité IT** doit être créé, contenant :
— Les risques
— Les contrôles recommandés
— La priorité d'action pour chaque risque
— Les contrôles sélectionnés sur base d'une analyse coût/bénéfice
— Les ressources nécessaires pour l'implémentation de ces contrôles
— Le personnel responsable
— Les dates de débuts et de fin d'implémentation
— Les requirements pour la maintenance, et tout autre commentaire

Voici un exemple de plan d'implémentation :

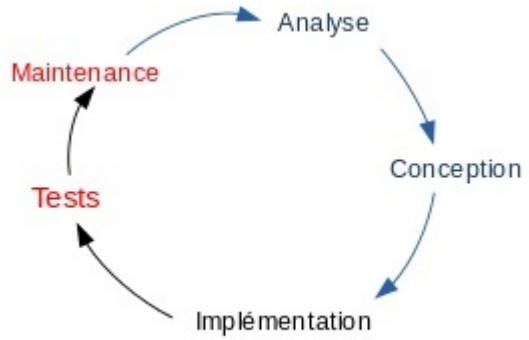
Risk (Asset/Threat)	Hacker attack on Internet router
Level of Risk	High
Recommended Controls	•Disable external telnet access •Use detailed auditing of privileged command use •Set policy for strong admin passwords •Set backup strategy for router configuration file •Set change control policy for the router configuration
Priority	High
Selected Controls	•Strengthen access authentication •Install intrusion detection software
Required Resources	•1 day of training for network administration staff
Responsible Persons	John Doe, Lead Network System Administrator, Corporate IT Support Team
Start – End Date	1-Feb-2011 to 4-Feb-2011
Other Comments	•Need periodic test and review of configuration and policy use

5 Business Continuity Plan

C'est un plan permettant à une organisation de fonctionner même en cas de désastre (vol, incendie, innondation, ...), éventuellement en mode dégradé ou en situation de crise majeur.
Son objectif est de minimiser les impacts d'une crise ou d'une catastrophe naturelle, technologique ou sociale sur l'activité (et donc la pérennité) d'une entreprise, d'un gouvernement, d'une institution, d'un groupe...

Par définition, un plan de continuité ne peut être définitif. Il doit être mis à jour en fonction du contexte et/ou des retours d'expériences et d'exercices (quand les exercices existent, ce qui est recommandé par les gestionnaires de crise). Une certaine standardisation des protocoles de secours se met en place qui semble utile, mais ne doit pas freiner l'innovation et une certaine souplesse nécessaire face à l'imprévu. cfr Wikipedia

Voici une image représentant le cycle de vie du Business Continuity :



6 Disaster Recovery Plan

6.1 Définition

Permet d'assurer, en cas de crise majeure ou importante d'un centre informatique, la reconstruction de l'infrastructure et la remise en route des applications supportant l'activité d'une organisation.
Le **plan de reprise d'activité** doit permettre, en cas de sinistre, de basculer sur un système de relève capable de prendre en charge les *besoins informatiques* nécessaires à la survie de l'entreprise.

6.2 Concepts importants

1. **Recovery Time Objective (RTO)** : le délai de rétablissement d'un processus, suite à un incident majeur, pour éviter des conséquences importantes associées à une rupture de la continuité d'activité. Il définit le temps alloué pour faire le basculement vers le nouveau système.
2. **Recovery Point Objective (RPO)** : Le RPO commence à s'exprimer à l'instant où l'incident majeur arrive et peut être exprimé en secondes, minutes, heures ou jours. Il s'agit donc de la quantité maximale acceptable de perte de données. C'est la durée des fichiers ou des données dans le stockage de secours exigé par l'organisation pour reprendre des opérations normales après l'incident. Ce critère définit l'état dans lequel doit se trouver le nouveau système après basculement.

7 Sécurité physique et des infrastructures

7.1 Objectifs

— Empêcher les dommages à l'infrastructure physique qui sous-temps le système d'information (HW, réseau, bâtiments, systèmes de contrôle de l'environnement, ...)
— Empêcher les mauvaises utilisations (accidentelles ou non) de l'infrastructure qui pourraient nuire à l'information (vol, copie, accès non autorisé, ...)

7.2 Menaces

1. **Menaces environnementales** qui ont pour cause des désastres naturels comme une tornade, un tremblement de terre, orages, inondations, ... Ici, les réels menaces sont entre autre la température ou taux d'humidité inadéquats, feu et fumée, dégâts des eaux, nuisances chimiques/radiologiques/biologiques, la poussière, infestations, ...
2. **Menaces techniques**. On pense notamment à l'alimentation électrique et ses problèmes de sous-voltages, sur-voltage dû à la foudre ou le bruit. Ou alors aux interférences électromagnétiques causés, par exemple, par du bruit sur une ligne d'alimentation électrique.
3. **Menaces liées aux personnes** :
   — Accès non autorisé
   — Vol
   — Vandalisme
   — Mauvaise utilisation des ressources

8 Sécurité et ressources humaines

8.1 Objectifs

— Améliorer le comportement des employés
— Accroître la capacité à tracer les actions des employés
— Diminuer la dépendance de l'organisation au comportement d'un employé
— Se conformer aux obligations contractuelles et de régulation

Afin d'arriver à ces objectifs, quelques efforts sont à effectuer. Ces derniers sont les suivants :

1. **Conscientisation** :
   — les employés sont conscients de leur responsabilité au niveau de la sécurité et des restrictions sur leurs actions, et agissent en fonction
   — les utilisateurs comprennent l'importance de la sécurité pour le bien-être de l'organisation
2. **Entrainement** :
   — Donner au personnel les compétences pour pouvoir mener à bien leurs tâches IT de manière plus sécurisé (Quoi et Comment)
   — **Utilisateurs généraux** : Fermer les portes, utiliser les mécanismes d'authentification, signaler les anomalies de sécurité
   — **Développeurs et spécialistes systèmes** : Intégrer la sécurité dans les cycles de vie des produits, apprendre comment limiter les vulnérabilités et comment surveiller les systèmes

— **Gestionnaires** : Doivent pouvoir faire des compromis entre les risques, coûts et bénéfices impliquant la sécurité

3. **Education** :
   — Formation des spécialités en sécurité
   — Formation généralement en dehors des organisations (certifications, formations universitaires, ...)

9 Pratiques et politiques d'emploi

Les employés d'une entreprise peuvent être impliqués dans des violations de sécurité de 2 manières :
— Aide involontaire
— Violation volontaire de la politique de sécurité

A ceci sera lié diverses **menaces** telles que l'accès non autorisé, altération de données, suppression de backups, destruction de systèmes, ...

Afin de minimiser ce type de problèmes, il est intéressant de prendre quelques précautions lors de l'embauche, pendant l'emploi et en fin de contrat d'un employé.

1. **Au moment de l'embauche** :
   — Vérification des CVs
   — Engagement vis-à-vis des procédures de sécurité
2. **Pendant l'emploi** :
   — Principe du *Least Privilege*
   — Séparation des responsabilités de sécurité
   — Ne pas dépendre d'employés-clés (départs, maladie, ...)
3. **En fin de contrat** :
   — Supprimer les accès de la personne virée
   — Informer les gardes du départ de la personne
   — Changer les codes des serrures, les cartes d'accès au système, voire les serrures physiques. Et Oui tout ça à cause d'un seul et unique connard de merde
   — Récupérer tous les avoirs à disposition de la personne (disque, ID, documents, équipements, ...)
   — Informer tous les département de l'entreprise du départ du gros connard

10 Politiques eMail et Internet

Définition des règles d'accès et d'utilisation des emails et du web :
— Business Use Only ?
— Accès au mail en extérieur ?
— Accès au mail privé
— Standards de conduite dans l'utilisation des ressources
— Propriété des contenus
— Actions disciplinaires
— Empêcher les personnes comme Sergen d'aller voir des sites plus que douteux avec Cortex les pyramides qui b**** une s****** française. (Celui-ci c'est pour rire, je précise on sait jamais)