Sécurité des réseaux

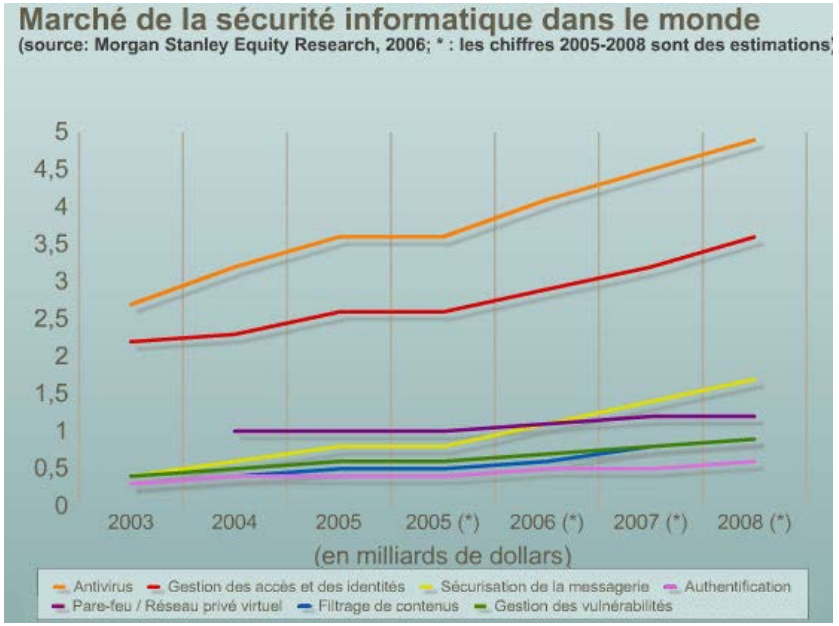
Bruno MARTIN

Laboratoire I3S

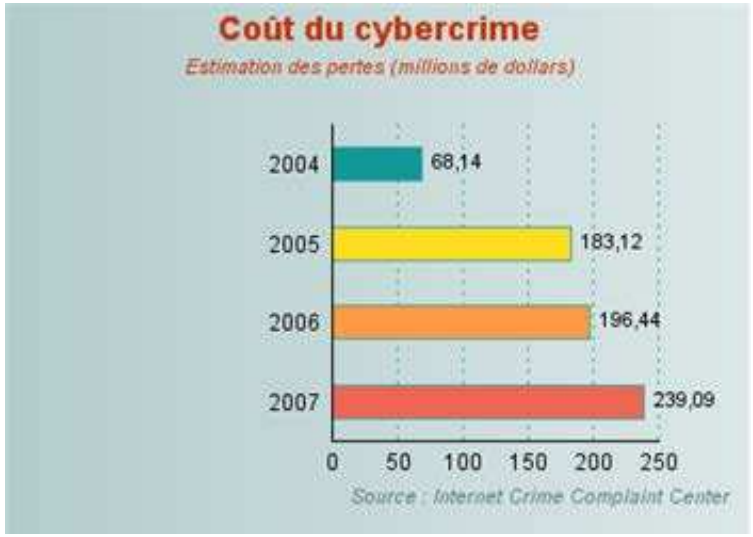
Département informatique

Université Côte d'Azur

Quelques chiffres



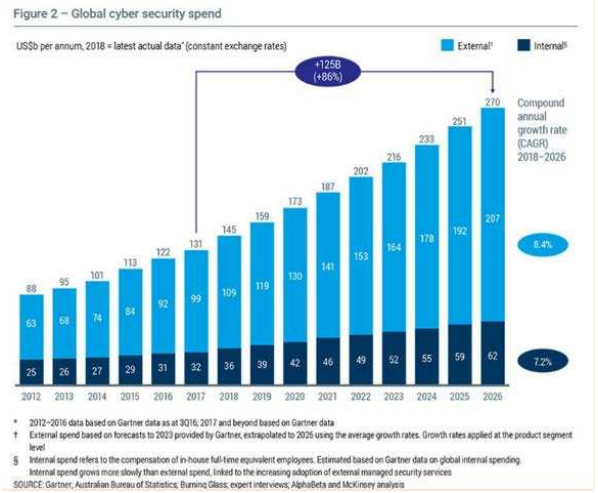
Quelques chiffres



327 milliards d'euros en 2014. 445 milliards de dollars en 2015 (budget de la France). Selon Juniper Research Report 2,8 milliards d'euros en 2017 pour les seules failles de sécurité.

Revue factuelle

Dépenses cybersécurité :



Quelques autres chiffres : 2017 et 2019.

Quelques chiffres

Statistiques : erreurs informatiques en 16 ans [CLUSIF-APSAD].

	Pertes en M€		
Origine	1984	1994	2000
Facteur humain	309	280	177
Erreurs	269	426	338
Fraude	335	998	???

80% des pertes dûes à des fraudes faites par des employés.

IBM (web) : 5 chiffres et voir la sécurité en quelques chiffres

Quelques chiffres

2 à 10$ prix moyen de la vente de numéros de cartes bancaires selon les payes et les plafonds

5$ tarif de location d'1h de botnet pour saturer un site Internet

2399 $ le tarif du malware Citadel permettant d'intercepter des numéros de cartes bancaires (et un abonnement mensuel de 125 $)

Source : Cyberedu

Vu dans la presse

Vu dans la presse

Selon un rapport gouvernemental britannique, élaboré avec le concours de la société de conseil PricewaterhouseCoopers (PwC) et publié à l'occasion de l'édition européenne du salon spécialisé Infosecurity, les entreprises d'outre-Manche ont – en moyenne – multiplié par trois les sommes consacrées à la sécurité informatique en l'espace de six ans. Une société britannique emploie désormais 7 % de son budget informatique pour la sécurité, contre 2 % en 2002.

De fait, 90 % des entreprises disent effectuer une copie de sauvegarde de leurs systèmes, avoir mis en place des filtres antispam, des pare-feu, des antivirus et des antispywares. 55 % ont une politique de sécurité documentée, contre 27 % en 2002. Et 40 % forment leurs salariés à de bonnes pratiques de sécurité, un chiffre qui a doublé. Résultat : le coût total des failles de sécurité a chuté de 35 %.

Cependant tout est encore loin d'être parfait. Un quart des sociétés britanniques a, malgré tous ces efforts, constaté de sérieuses failles de sécurité au cours des deux dernières années. 21 % d'entre elles consacrent moins de 1 % de leur budget à se protéger du piratage. Et persistent des pratiques informatiques dignes de l'ère pré-Internet. Ainsi, 35 % des entreprises britanniques ne contrôlent pas l'usage de la messagerie instantanée. Et 84 % ne vérifient jamais si les courriels sortants contiennent des informations confidentielles.

Vu au cinéma

Plan

Motivation & risques

- ↑ échanges Internet :
  - informations
  - commerciales
- ↑ risques :
  - fraudes diverses
  - piratage

$\Rightarrow$

- interception messages
  - « bris » de mdp
  - vol d'infos en transit
- intrusion des systèmes
  - vol d'infos mémorisées
  - virus
  - détournement de biens
- faux clients, escroquerie

Tant en interne qu'en externe

Buts de la sécurité

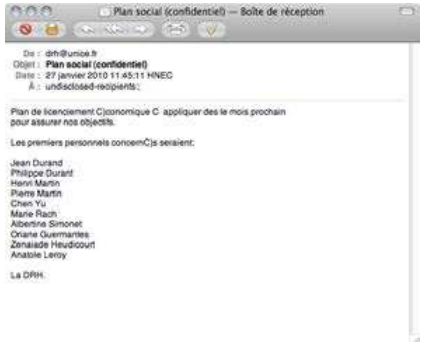
Améliorer la sécurité face aux risques identifiés. Pouvoir assurer :

- **disponibilité :** aptitude du système à remplir une fonction dans des conditions prédéfinies d'horaires, de délai ou de perf.
- **intégrité :** garantit que l'information n'est pas modifiée sauf par une action volontaire et autorisée ;
- **confidentialité :** l'information n'est seulement accessible qu'à ceux dont l'accès est autorisé
- **authentification :** preuve de l'identité d'une personne ou d'un système
- **prouvabilité :** possibilité de reconstituer un traitement pour le contrôle ou la preuve.

2 types de sécurité :

- **Sécurité des données :** à l'intérieur d'un système ; (cryptographie et théorie des codes)
- **Sécurité des réseaux :** données qui transitent entre des systèmes, (environnement distribué ou réseau).

Illustration



- L'information est publique ?
- Provient-elle de la DRH ?
- A-t-elle été modifiée ?

Sécuriser le mail par PGP : TP4

Panorama des risques informatiques

Classification par le CLUSIF [http://www.clusif.fr/] basées sur les délarations de sinistres des entreprises :

- accidents naturels : incendie, dégâts des eaux, etc.
- perte des services essentiels : coupure courant, réseau, rupture de stocks
- erreurs : tous les stades de l'activité : analyse, conception, réalisation, mise en œuvre, utilisation
- malveillance : vol, vandalisme, fuite d'informations

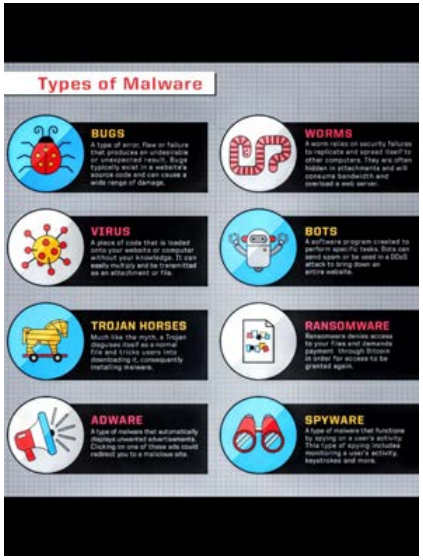
Panorama des menaces

- Malware : ransomware, cheval de troie, enregistreur de frappe, virus, spyware, vers
- Messagerie : canulars, chaines, phishing, spam
- Web : cookies, faux sites (web 2.0 devient applicatif)
- Smartphones : de plus en plus attaqués
- Spoofing : changement de pages d'accueil
- Social engineering : fuite de données

Techniques : Attaque en force, par déni de service, botnet, correctifs, buffer overflow, exploitations

Ces menaces peuvent être combinées dans une même attaque

Exemples de malwares



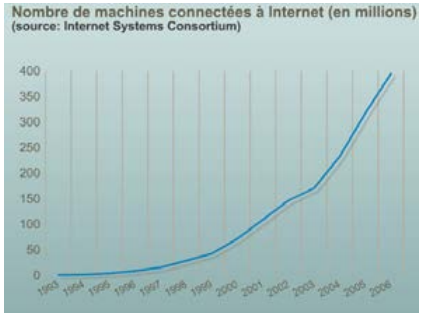
Sans oublier Dave...

Premier constat

Internet (1969...) pas conçu pour la sécurité :

- peu d'utilisateurs
- pas de besoin

Depuis : Vie privée ; nx acteurs, services, usages : Internet accroît son audience et son influence.



Types d'attaques & menaces

- **passives :**
  - observation non autorisée
  - accès non autorisé à de l'information
- **actives :**
  - contrôle non autorisé d'un système
  - modif de l'information
  - accès à des services
  - refus de service aux utilisateurs légaux
- **intrusion :** de toute provenance (réseau, terminal local, programme)
- **refus de service :** atteinte à la disponibilité. Conséquence classique des virus ou des ping of death, low ion orbit canon
- **vol d'informations :** pas nécessaire de pénétrer un système. Une attaque passive peut suffire (login).

Qui attaque ?

- les gouvernements :
  - NSA aux états unis
  - groupes Advanced Persistent Threats (Chine, Russie, Iran,...)
  - DGSE, DGSI en france
- le crime organisé
- les concurrents

Types d'attaquants

- **Le hacker "canal historique" :** pour le prestige, améliorer la qualité des logiciels (espèce en voie de disparition)
- **le "hacktiviste" :** pour faire passer un message politique (ex. anonymous)
- **le cyber-délinquant :** pour gagner de l'argent (espèce en forte croissance, jusqu'aux organisations mafieuses)
- **le cyber terroriste :** pour marquer les esprits et déstabiliser avec des attaques importantes.
- **les états étrangers :** pour déstabiliser un état, paralyser les services essentiels, préparer une cyber-guerre
- **les cyber-mercenaires :** comparables aux cyber-terroristes mais agissant seul

Plan

Avertissement !

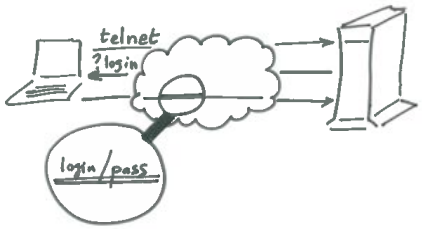
- L'utilisation des outils décrits plus loin est **illégale**
- interdiction FORMELLE de les utiliser ailleurs que dans les salles ou plate-formes prévues à cet usage
- outils qui servent autant à protéger qu'à attaquer
- ne JAMAIS les utiliser ailleurs que sur un LAN privé/virtuel
- charte informatique

Attention !

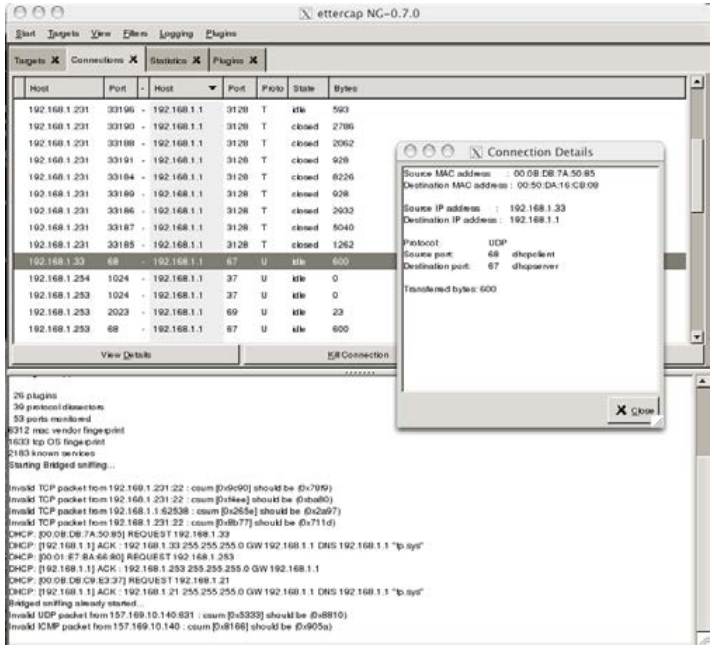
- Outils d'audit = outils d'attaque = armes
  - ne pas les pointer sur des cibles réelles
  - prendre toutes les précautions
  - demander l'autorisation de l'admin et du FAI
  - pas dans le cadre de la fac
- Perpétrer des actes de piratage est répréhensible
  - peut ruiner votre carrière
  - C'est **TRÈS** sérieux

Explication

- Attaque passive
- Vol d'information
- Malveillance
- Possible avec :
  - telnet
  - pop
  - imap
  - http
  - ...



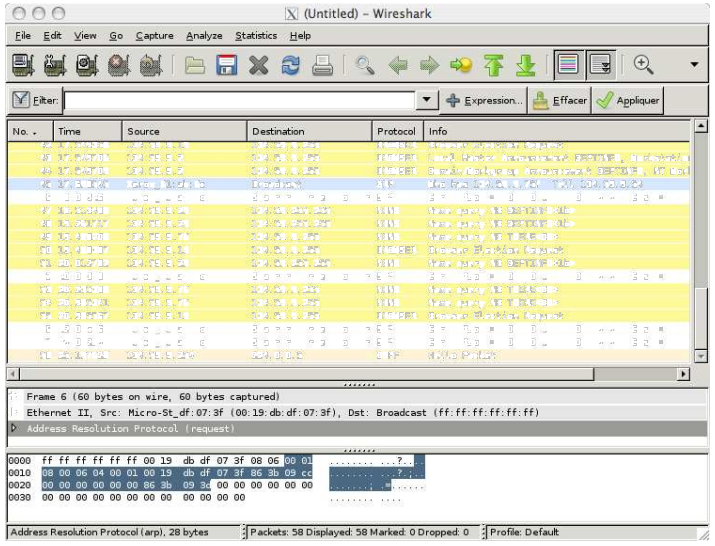
Ettercap



Ettercap- description

- suite for MIM attacks on LAN
- features sniffing of live connections, content filtering on the fly and many other interesting tricks
- supports active and passive dissection of many protocols (even ciphered ones) and includes many features for network and host analysis

Wireshark



Outils d'audit

Packet sniffers : logiciels d'écoute des données non-chiffrées d'un LAN. Servent à

- intercepter mdp (ou autre info) qui transite en clair
- résoudre des problèmes réseaux en visualisant le traffic
- la rétro-ingénierie réseau

`http://fr.wikipedia.org/wiki/Packet_sniffer`

Attention à l'utilisation selon l'architecture du LAN (hub ou switch) !

Rappels sur ethernet

L'acheminement des trames ethernet s'appuie sur la notion de compétition pour l'accès au media.

Chaque hôte écoute et attend un « silence media » avant d'émettre. Si un signal transite sur le media, les émetteurs attendent que le media soit libre.
Quand le canal est libre, un hôte émet.
Quand 2 hôtes émettent simultanément, il y a **collision**. Dans ce cas, les 2 hôtes détectent la collision et envoient un signal de collision qui empêche l'émission pour une durée aléatoire.

Un **domaine de collision** est une région du réseau au sein de laquelle les hôtes partagent l'accès au media.

Fonctionnement hub (concentrateur)

Quand un paquet est reçu, il est propagé sur toutes les interfaces sauf celle de l'émetteur. Hub ne délimite ni les domaines de collision ni les domaines de broadcast.

Permet à la carte réseau d'un hôte d'accepter tous les paquets qu'elle reçoit, même s'ils ne lui sont pas destinés (mode de promiscuité).

Détection du mode de promiscuité :

- augmentation charge de l'hôte qui traite tous les paquets et augmente la latence du réseau
- détection avec `detectpromisc`

Inconvénients hub

Réseau avec beaucoup d'hôtes, problèmes de performance :

- **disponibilité :** partage de la bande passante ; un hôte peut monopoliser tout le trafic (gros transfert)
- **latence :** (temps nécessaire à un paquet pour atteindre sa destination). Avec des hubs on attend une opportunité de transmission pour éviter les collisions. Latence croît en fonction du nombre d'hôtes du réseau.
- **défaillance :** plus sensible aux pannes ou aux mauvaises configurations de la vitesse de transmission

Un switch (commutateur) résout ces problèmes.

Avantages switch (commutateur)

- Switch améliore la disponibilité et la latence en délimitant les domaines de collision
- Chaque hôte connecté dispose de toute la bande passante.
- un paquet qui arrive sur un port du switch n'est retransmis que sur le port auquel le destinataire est connecté
- Signaux de collision non retransmis par les switches

Fonctionnement switch

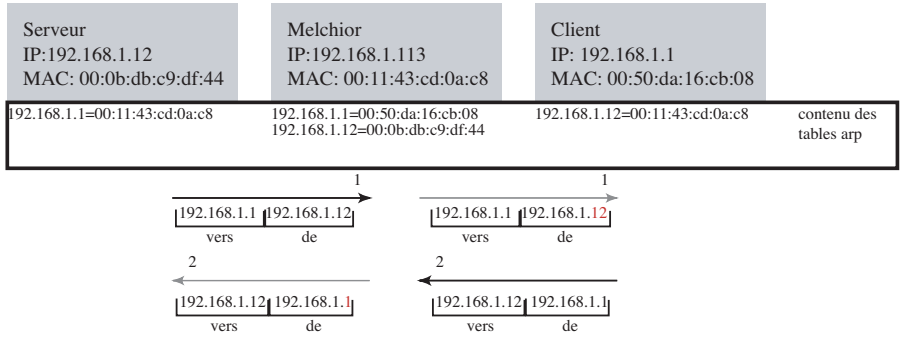
Un paquet qui arrive dans le switch est mis dans le buffer.
L'adresse MAC du paquet est lue et comparée à la liste des MAC connues rangées dans la **table de lookup**.
3 modes d'acheminement :

- **cut through :** lecture des 6 octets MAC dest. et routage direct sans traitement vers le port du destinataire
- **store and forward :** mise en mémoire et traitement du paquet avant son acheminement (rejeter les paquets mal formés, gérer les messages de collision)
- **fragment free :** analogue au cut through mais lit les 64 premiers octets avant le routage. (cela limite les erreurs de collision qui arrivent souvent sur les 64 premiers octets)

Le mode le plus utlisé est le store and forward.

Ecoute sur réseau switché

Mode promiscuité inutile. Utiliser arp spoofing/poisoning :



2 outils : `dsniff` et `ettercap` disponibles sur Kali. Nécessaires pour utiliser des outils d'écoute du réseau (`dsniff`, `wireshark`, `tcpdump`).

Utilisation `dsniff`

Contient un utilitaire `arpspoof`.
Pour mener à bien l'attaque :

- activer l'**IP forwarding** sur M ; l'IP forwarding permet de faire transiter des paquets d'une interface réseau à une autre. La machine va servir de « routeur »
- activer le spoofing dans les 2 sens :

```
arpspoof -t IP1 IP2 & >/dev/null
arpspoof -t IP2 IP1 & >/dev/null
```

- terminer par :

```
killall arpspoof
```

Utilisation `ettercap`

`ettercap` a 2 modes de fonctionnement : interactif (interface ncurses ou Gtk) ou en CLI.
Pour mener à bien l'attaque :

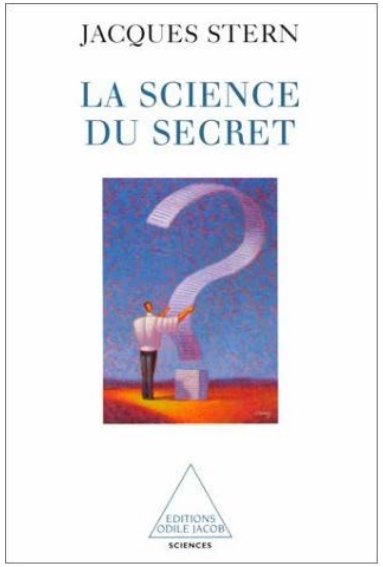
- l'IP forwarding est automatiquement activé par `ettercap`
- empoisonner tout le trafic par :
  `ettercap -T -q -M ARP // //`
  - `-T` : choix type interface
  - `-q` : mode silencieux (quiet)
  - `-M ARP` : attaque MIM type `arp`
  - `// //` : de la source vers la destination
- empoisonner une cible (`IP1`) par :
  `ettercap -T -q -M ARP /IP1/ //`
  redirige tout le trafic entre `IP1` et le reste du réseau

Plan

Historique

3 âges de la crypto. pour J. Stern :

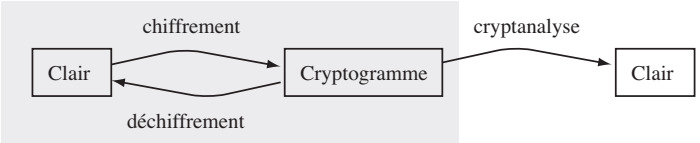
- artisanal : modifier l'écriture
- technique : machines à chiffrer
- paradoxal : cryptographie à clé publique



regardez-le : univ. de tous les savoirs

Cryptologie = cryptographie + cryptanalyse

science de la communication en présence d'adversaires.



- **chiffrer** un **clair**→ **cryptogramme** (**confidentialité**).
- Destinataire légitime **déchiffre** le cryptogramme → clair.
- **cryptanalyste** ne peut **décrypter** le cryptogramme.

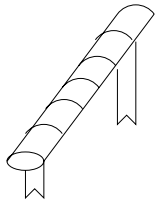
Âge artisanal – César

le clair `toute la gaule` devient `WRXWH OD JDXOH`.

(`A` devient `d`, `B` devient `e`...)

Substitution

Transposition : scytale



Transposition simple à tableau

A partir d'une phrase clé, on définit une clé numérique :

T	R	A	N	S	P	O	S	I	T	I	O	N	S	I	M	P	L	E
18	14	1	8	15	12	10	16	3	19	4	11	9	17	5	7	13	6	2

On chiffre, «le chiffrement est l'opération qui consiste à transformer un texte clair, ou libellé, en un autre texte inintelligible appelé texte chiffré ou cryptogramme» [2].

18	14	1	8	15	12	10	16	3	19	4	11	9	17	5	7	13	6	2
l	e	c	h	i	f	f	r	e	m	e	n	t	e	s	t	l	o	p
é	r	a	t	i	o	n	q	u	i	c	o	n	s	i	s	t	e	à
t	r	a	n	s	f	o	r	m	e	r	u	n	t	e	x	t	e	c
l	a	i	r	o	u	l	i	b	e	l	l	é	e	n	u	n	a	u
t	r	e	t	e	x	t	e	i	n	i	n	t	e	l	l	i	g	i
b	l	e	a	p	p	e	l	é	t	e	x	t	e	c	h	i	f	f
r	é	o	u	c	r	y	p	t	o	g	r	a	m	m	e			

On prend ensuite par blocs de 5 lettres les colonnes prises dans l'ordre défini par la clé.

Âge technique – Enigma

le clair `alles in ordnung` devient `EDCGZVRRIOVRAY`

Sécurité parfaite – Vernam 1917

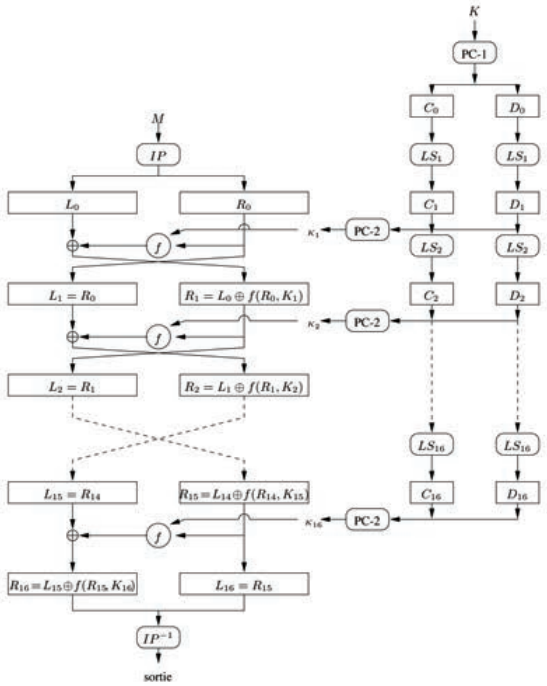
Ce one-time pad est-il un chiffre «parfait» ?

A et B partagent une suite aléatoire de n bits : la clé secrète K.
A chiffre M de n bits en $C = M \oplus K$.
B déchiffre C en $M = K \oplus C$.

Exemple

$M = 0011$, $K = 0101$
$C = 0011 \oplus 0101 = 0110$
$M = K \oplus C$.

Non-réutilisation : à chaque nouveau message, engendrer une nouvelle clé.



Chiffres produits et itérés [3]

Amélioration : combiner substitutions et transpositions.

Un chiffre est **itéré** si le chiffré est obtenu par applications itérées d'une fonction de tour. A chaque tour, on combine le texte d'entrée avec une clé de tour.

Définition

Dans un chiffre itéré à r tours, le chiffré est calculé par application itérée au clair d'une ***fonction de tour*** *g t.q.*

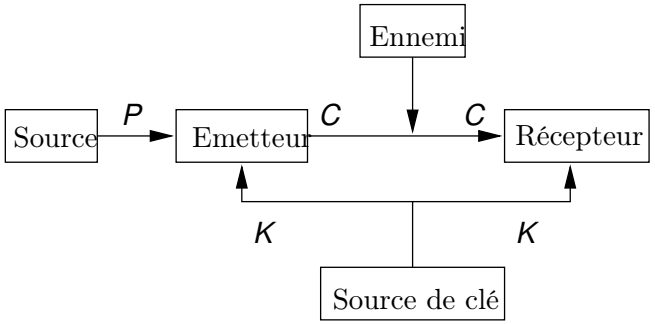
$$C_i = g(C_{i-1}, K_i) \quad i = 1, \ldots, r$$

où C_0 est le clair, K_i une clé de tour et C_r le chiffré. Déchiffrement en inversant l'équation précédente : pour une clé fixée K_i, g doit être inversible.

Le résultat

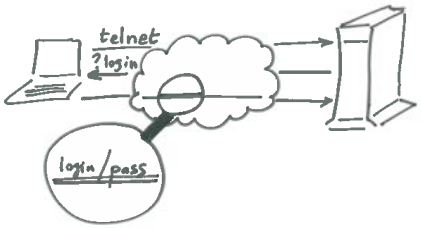
Chiffre à clé secrète

Modèle de Shannon pour le secret [3] :



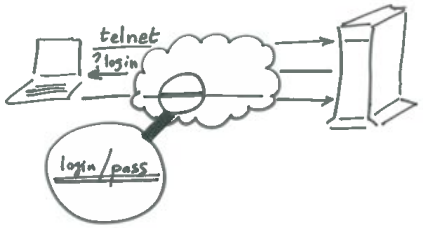
Retour à l'exemple

- Attaque passive
- Vol d'information
- Malveillance
- **Plus possible**



Retour à l'exemple

- Attaque passive
- Vol d'information
- Malveillance
- **Plus possible**
- Distribution des clés ?



Tentative

Pré-distribuer une clé à chaque couple d'utilisateurs dans OS...

Petit calcul :

- 2 utilisateurs : 2 clés
- 4 utilisateurs : 6 clés
- n utilisateurs : $\frac{1}{2}\binom{n}{2}$

Environ 3.10^6 machines connectées...
Nombre de clés....
Mémoire pour les stocker : ...
Evolution ?

Tentative

Pré-distribuer une clé à chaque couple d'utilisateurs dans OS...

Petit calcul :

- 2 utilisateurs : 2 clés
- 4 utilisateurs : 6 clés
- n utilisateurs : $\frac{1}{2}\binom{n}{2}$

Environ 3.10^6 machines connectées...
Nombre de clés....
Mémoire pour les stocker : ...
Evolution ?

Les engendrer et les transmettre

Tentative

Pré-distribuer une clé à chaque couple d'utilisateurs dans OS...

Petit calcul :

- 2 utilisateurs : 2 clés
- 4 utilisateurs : 6 clés
- n utilisateurs : $\frac{1}{2}\binom{n}{2}$

Environ 3.10^6 machines connectées...
Nombre de clés....
Mémoire pour les stocker : ...
Evolution ?

Les engendrer et les transmettre
Comment ?

Protocole de Diffie-Hellman

Procédé qui permet d'établir une clé partagée entre plusieurs entités de telle sorte qu'aucune d'entre elle ne puisse établir sa valeur par avance.
On cherche une solution qui permet à deux entités :

- qui ne se sont jamais rencontrés
- qui ne possèdent pas d'information partagée

de construire une clé secrète commune

- connue d'eux seuls
- inconnue de quiconque, même d'un indiscret qui écouterait leurs communications.

L'idée
Imaginer une solution facile à calculer pour les utilisateurs légaux et difficile pour un indiscret : fonction **à sens unique**.

Mise en accord par Diffie Hellman [5]

- **Etape préliminaire**
  - On choisit q un grand premier
  - On choisit a, $1 < a < q$
- **Les clés :** Chaque utilisateur U :
  - choisit aléatoirement X_U, $1 < X_U < q$ conservée secrète
  - publie $Y_U = a^{X_U} \mod q$

A et B construisent une clé commune avec : Y_A et Y_B.

- A calcule $K = Y_B^{X_A} \mod q$
- B calcule $K = Y_A^{X_B} \mod q$

A et B ont alors une clé (secrète) commune K :

$$Y_B^{X_A} \equiv (a^{X_B})^{X_A} \equiv a^{X_B X_A} \equiv Y_A^{X_B} \mod q$$

Chiffres à clé publique

Invention de Diffie et Hellman [1] ; phrase prophétique :

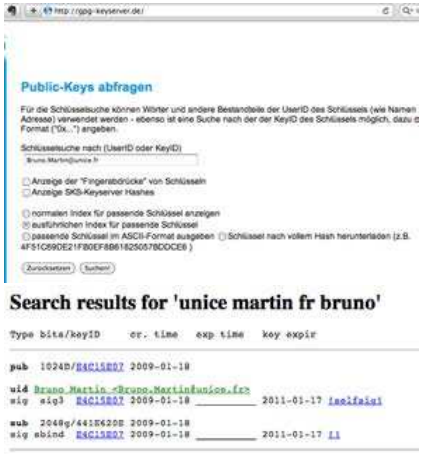
> *Nous sommes aujourd'hui à l'aube d'une révolution en cryptographie.*

Idée géniale : asymétrique ; chiffrement $\neq$ du déchiffrement.

- chiffrement par clé **publique**.
- déchiffrement avec clé **privée**.

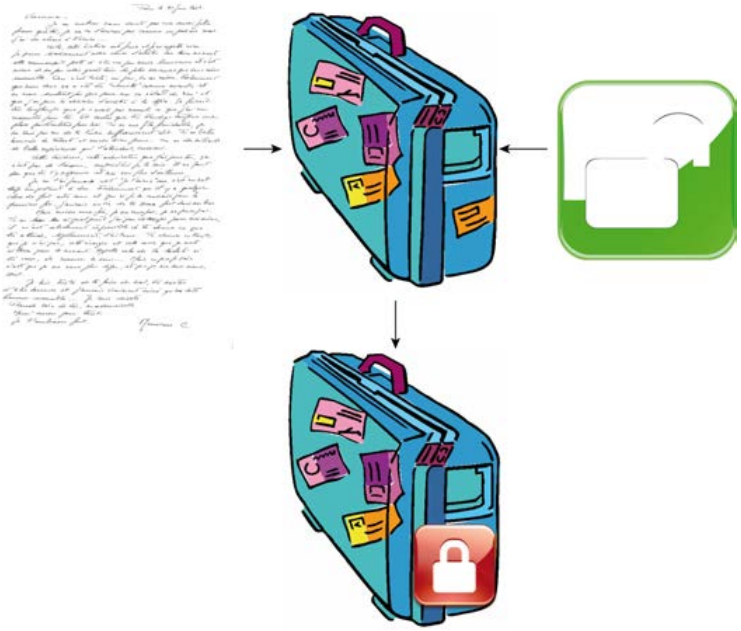
Les clés sont... publiques

Enfin, celles qui servent à chiffrer ; elles sont dans un annuaire :

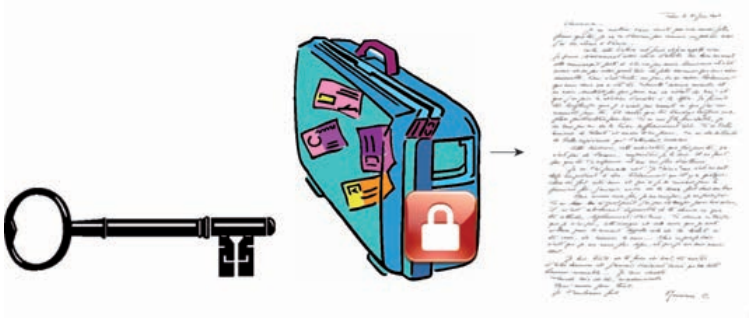


`Bruno.Martin@unice.fr`

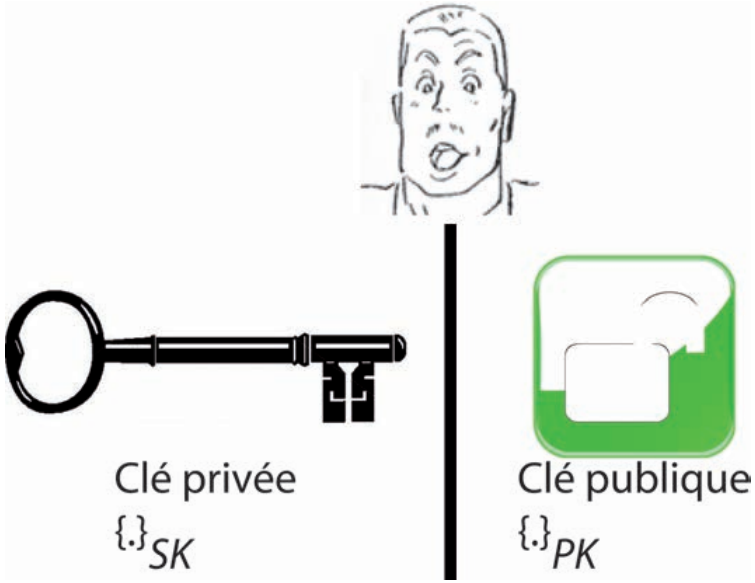
Envoi d'un message



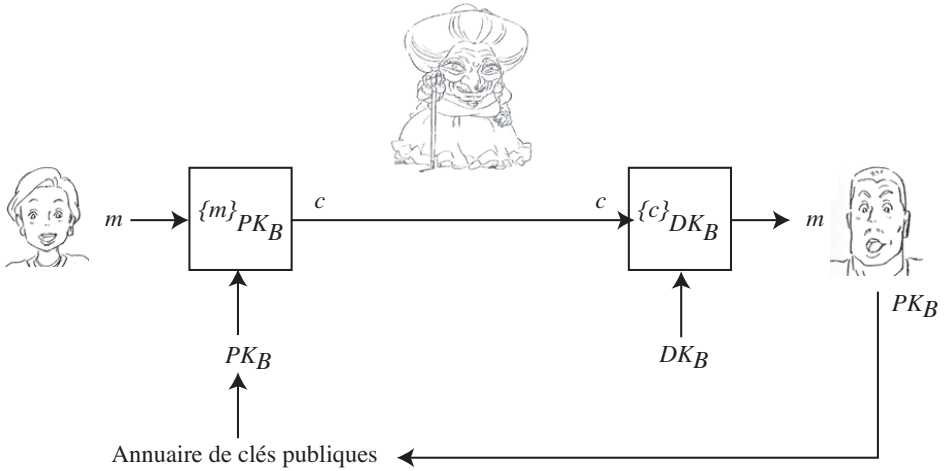
Réception d'un message



Paire de clés

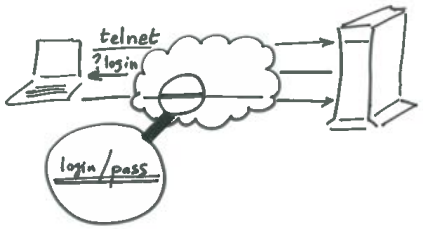


Chiffre à clé publique



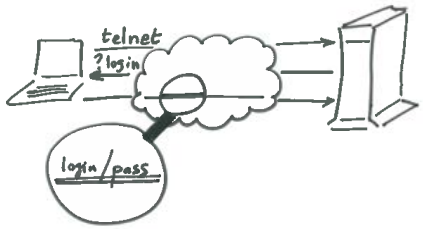
Retour à l'exemple

- Attaque passive
- Vol d'information
- Malveillance
- **Plus possible**



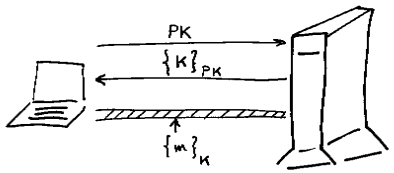
Retour à l'exemple

- Attaque passive
- Vol d'information
- Malveillance
- **Plus possible**
- Distribution clés par PKC

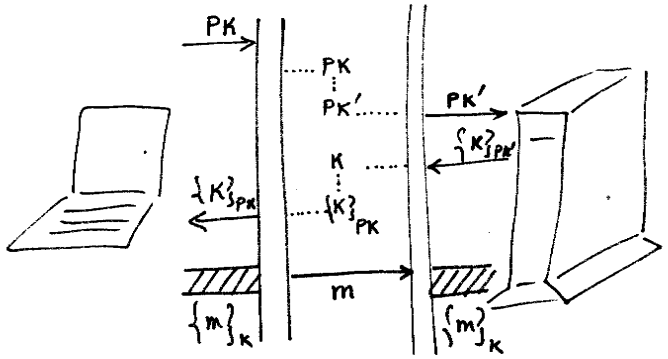


La solution ?

- $\{.\}_K$ pour chiffrer
- K transmis avec $\{K\}_{PK}$
- On a fini ?



Nouvelle attaque !



Man In the Middle

Pourquoi le chiffrement hybride ?

A disadvantage of asymmetric ciphers over symmetric ciphers is that they tend to be about "1000 times slower." By that, I mean that it can take about 1000 times more CPU time to process an asymmetric encryption or decryption than a symmetric encryption or decryption.

Synthèse

Le chiffrement

- garantit la confidentialité
- pas l'authentification
- d'autres attaques possibles

Empêcher l'attaque MIM...

Assurer l'authentification...

Avec les signatures

Objectif des signatures

- Seul l'expéditeur doit pouvoir signer
- N'importe qui peut vérifier la signature
- La signature dépend uniquement :
  - de l'identité de l'expéditeur
  - du message
- Garantit :
  - authentification de l'expéditeur
  - integrité du message

Signature

Principe : échanger les rôles de *pk* et de *sk*

- algo. signature (privé) noté **sig** qui, pour une clé fixée *sk*, retourne une signature *S* pour un message *m* ;

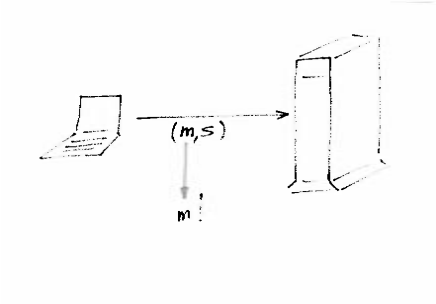
$$\text{sig}_{sk}(m) = \{m\}_{sk} = s$$

- vérification noté **ver** qui, à une clé fixée *pk* et pour tout couple message/signature (m, s) va vérifier si la signature correspond bien au message.

$$\text{ver}_{pk}(m, s) = \begin{cases} \text{vrai si } s = \text{sig}_{sk}(m) \Leftrightarrow \{s\}_{pk} = m \\ \text{faux si } s \neq \text{sig}_{sk}(m) \end{cases}$$

Inconvénients

- On « voit » *m*
- Perte de confidentialité
- Taille *s* $\propto$ taille *m*



Diminuer la taille de la signature...

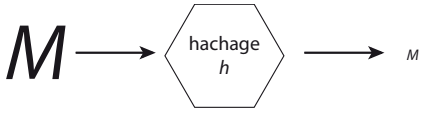
Utiliser le hachage

Hachage

h calcule

$$z = h(M)$$

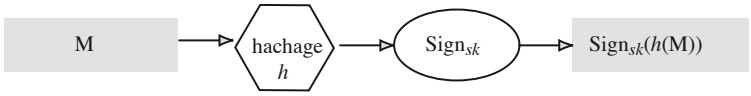
M de taille arbitraire,
z empreinte de taille fixe.



h est **à sens unique**, i.e.

- $h(M)$ rapide à calculer
- *z* difficile à inverser.

Signature avec hachage



Bob envoie M signé.
Clés : (pk, sk)
Il calcule

1. $h(M)$
2. $s = \{h(M)\}_{sk}$

Il envoie (M, s).

N'importe qui :

- reçoit (M, s)
- récupère pk de Bob
- vérifie :
  1. $z = h(M)$
  2. s signe $M \Leftrightarrow z = \{s\}_{pk}$

h connue de tout le monde (~~`md5`~~, `SHA-1`, `SHA-3` aka Keccak).

Rappel

On voulait contrer l'attaque MIM et assurer l'authentification.

But :

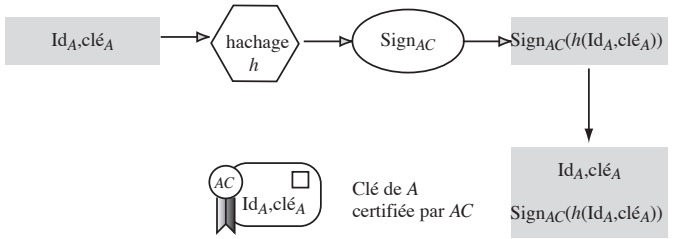
- transmettre une clé publique pk
- garantir la relation entre Id et pk

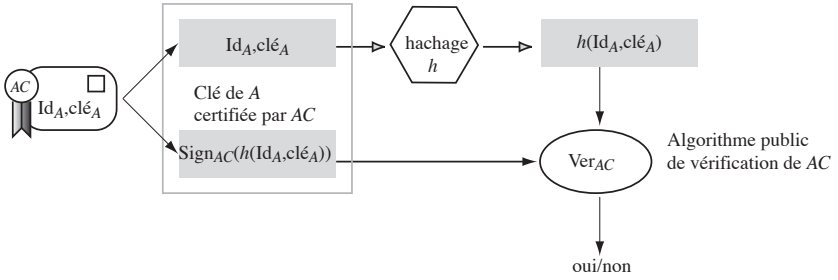
Comment faire ?

Principe

Une autorité va **garantir** la relation (Id, pk) [4]

Certification & Vérification





Paradoxe

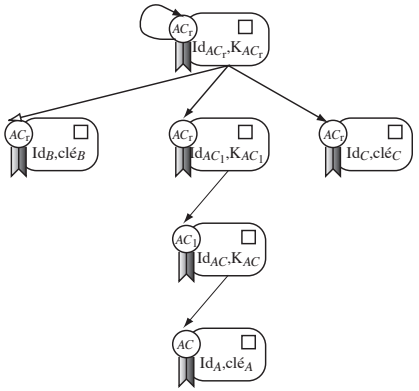
Comment connaît-on l'algorithme de vérification de l'autorité de certification ?

Quel est le modèle de confiance ?

- confiance directe
- confiance hiérarchique (le plus utilisé)
- toile de confiance (web of trust)

Chaîne de certification

Une AC certifie une autre AC. Bob remonte une chaîne de certification jusqu'à une AC en qui il a confiance.



Mauvaise AC : attaques possibles (2011 : DigiNotar)

Création d'une AC « racine »

Problème : il faut une AC « racine ».
Certificat auto-délivré. L'entité qui délivre le certificat est identique au sujet certifié.

- confiance : large distribution de la clé publique de l'AC.
- possible de se déclarer comme AC « racine »

Toile de confiance

Tentative de concilier les modes directs et hiérarchiques.
La confiance est accordée par l'utilisateur selon le principe que plus on dispose d'information, meilleure est la confiance. On accorde sa confiance directement ou au moyen d'une chaîne de certification qui remonte jusqu'à un tiers connu selon le principe :

- Jean signe l'identifiant Paul dans le certificat de Paul ;
- lorsque Marie veut vérifier que Jean a signé Paul, elle utilise la clé publique de Jean pour vérifier sa signature dans le certificat de Paul ;
- Marie fait confiance à Jean. Elle reçoit un message signé de Paul. Elle vérifie la validité de la signature de Paul (clé récupérée depuis un serveur de clés). Comme elle fait pleinement confiance à Jean, elle valide de plus le fait que c'est bien Paul qui a signé ce message.

C'est le système utilisé par `OpenPGP`, `GnuPG` ou `PGP`.

Rupture dans la chaîne hierarchique

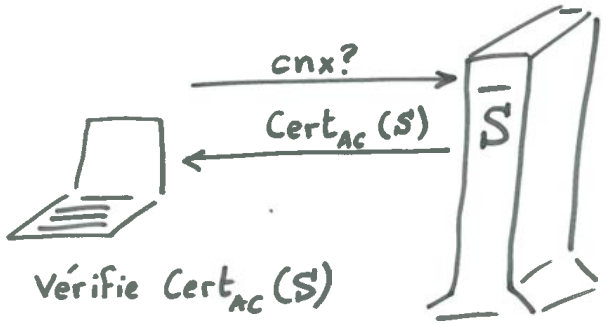
Une attaque MIM de plus haut niveau

Porte sur la transmission d'un certificat dont l'AC n'est pas connue. C'est le cas, par exemple, d'un certificat "auto-délivré".

Contenu du TP3

Certificat original et celui falsifié

Transmettre une clé publique



Rapide synthèse

Ce qu'on sait faire pour le moment :

- Transmettre une clé publique
- Transmettre une clé secrète
- Sécuriser un canal

Encore une attaque

- DOS
- empêcher la communication
- transmettre des $\text{Cert}_{AC}(S)$ erronés
- client passe son temps à faire des vérifications inutiles

Comment s'assurer de l'identité de S

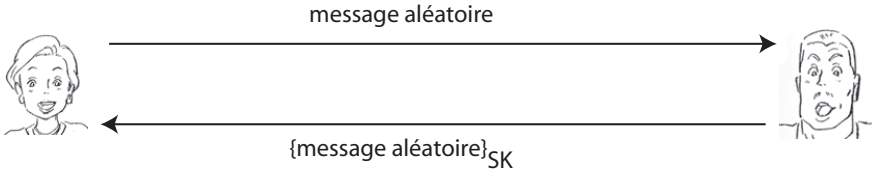
Identification et authentification

- **identification** affirmation de l'identité d'une entité : "je suis Bruno"
- **authentification** : vérifie l'identité d'une entité "je peux prouver que je suis Bruno"

Service d'authentification vérifie l'identité à différents niveaux :

- applicatif : `http, ftp`
- transport : `ssl, ssh`
- réseau : `ipsec`
- transmission : `pap, chap` (qui utilisent `md5`)

Exemple d'authentification « asymétrique »



- Sans calcul h(msg-aléa), KPA possible : $(m/\{m\}_{SK})$ connus, ...
- **Hyp :** Alice connaît la clé publique de Bob au préalable.

Plan

Identification–handshake

C'est le moyen pour Alice de vérifier l'identité de Bob.
pk_B la clé publique de Bob et sk_B sa clé privée.

$A \rightarrow B$	$r =$ un message aléatoire
$B \rightarrow A$	$c = \{r\}_{sk_B}$

Signer un message aléatoire r fourni par un tiers et le réexpédier peut s'avérer dangereux.
On pourrait utiliser une fonction de hachage h afin que Bob signe $h(r)$. Mais le danger persiste.

Identification–handshake

Mieux vaut que Bob signe un message de son cru

$A \rightarrow B$	*"Bonjour, est-ce Bob ?"*
$B \rightarrow A$	$m =$ *"Alice, je suis bien Bob"*
	$c = \{h(m)\}_{sk_B}$

Authentification–handshake

Alice n'a pas forcément déjà connaissance de la clé publique de Bob. Comment informer sûrement quelqu'un de sa clé publique ?

$A \rightarrow B$	*"Bonjour"*
$B \rightarrow A$	*"Bonjour, je suis Bob. Voici ma clé publique"* pk_B
$A \rightarrow B$	*"Prouve-le."*
$B \rightarrow A$	$m =$ *"Alice, c'est bien Bob"*
	$c = \{h(m)\}_{sk_B}$

N'importe qui peut se faire passer pour Bob aux yeux d'Alice, par une attaque MIM.

Transmettre un certificat–handshake

Certificat garantit la relation entre une identité et clé publique.

$A \to B$	*"Bonjour"*
$B \to A$	*"Bonjour, je suis Bob. Voici mon certificat"* $cert_B$
$A \to B$	*"Prouve-le."*
$B \to A$	$m =$ *"Alice, c'est bien Bob"*
	$c = \{h(m)\}_{sk_B}$

Eve pourrait se substituer à Bob dans les premiers échanges, mais échouera au dernier.

Echanger un secret–handshake

La communication par clés publiques est coûteuse, une fois finie la phase d'authentification, on échange une clé symétrique.

$A \to B$	*"Bonjour"*
$B \to A$	*"Bonjour, je suis Bob. Voici mon certificat"* $cert_B$
$A \to B$	*"Prouve-le".*
$B \to A$	$m =$ *"Alice, c'est bien Bob"*
	$c = \{h(m)\}_{sk_B}$
$A \to B$	*"Ok Bob, voici notre secret :"*
	$s = \{secret\}_{pk_B}$
$B \to A$	$m' = \{$message de Bob$\}_{secret}$

Attaque MIM

L'homme du milieu Melchior peut s'interposer dans les 5 premiers échanges. Arrivé au sixième, il peut brouiller le message de Bob, quitte à ne pas envoyer un message très intelligible à Alice :

$B \to M$	$m' = \{$message de Bob$\}_{secret}$
$M \to A$	altération de m'

Alice n'a aucune certitude quant à l'existence de Melchior, même si elle trouve suspect le dernier message de Bob.

SSL–handshake

Pour éviter cette incertitude, mieux vaut utiliser un MAC :

$M = h($un message de Bob, *secret*$)$

$A \to B$	*"Bonjour"*
$B \to A$	*"Bonjour, je suis Bob. Voici mon certificat"* $cert_B$
$A \to B$	*"Prouve-le".*
$B \to A$	$m =$ *"Alice, c'est bien Bob"*
	$c = \{h(m)\}_{sk_B}$
$A \to B$	*"Ok Bob, voici notre secret :"*
	$s = \{secret\}_{pk_B}$
$B \to A$	$m' = \{$message de Bob$\}_{secret}$
	$M = h($message de Bob, *secret*$)$

Melchior peut pertuber ce qu'il veut, M aura au moins l'avantage d'en avertir le destinataire.

La communication–record protocol

Ce protocole transmet un message de taille arbitraire. Il le découpe en blocs, le comprime éventuellement, ajoute un MAC, chiffre et transmet le résultat en ajoutant un numéro de séquence pour détecter s'il manque des messages ou si certains ont été altérés.

Il assure :

- confidentialité des données transmises par l'application
- intégrité des données
- authentification de l'origine

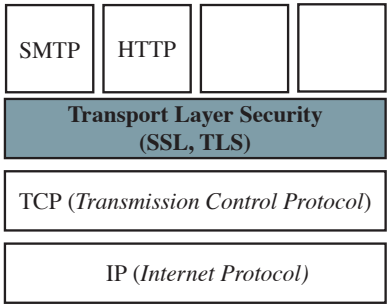
Une fois le record protocol achevé, les données chiffrées sont fournies à `TCP`.

Sécurité de TCP

Protocoles pour sécuriser TCP :

- **Secure Socket Layer**
- **Transport Layer Security** standardisé par l'IETF

Configuration TP2



Et, plus récemment, https://www.libressl.org, https://boringssl.googlesource.com/boringssl/ ou https://github.com/awslabs/s2n

Protocole vs Implementation

Ne pas confondre le protocole avec son implémentation ! TLS (qui remplace maintenant SSL) est implémenté par de nombreuses libraires. Voir à ce sujet https://en.wikipedia.org/wiki/Comparison_of_TLS_implementations. Les plus classiques :

- OpenSSL
- LibreSSL
- BoringSSL
- GnuTLS

Et voir aussi les recommandations sur Crypto Best Practice

Synthèse

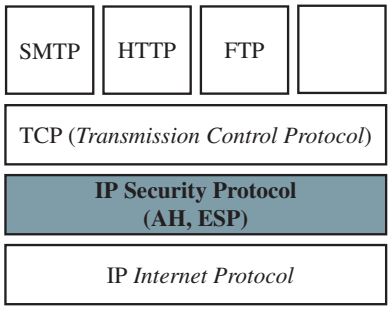
- SSL est un exemple de **protocole** : ensemble de règles permettant d'établir une communication entre deux entités
- fournit certains **services** de sécurité
  - identification, authentification
  - confidentialité
  - intégrité
- mis en place par les **mécanismes** de sécurité
  - chiffrement, signature
  - contrôle d'accès
  - hachage
  - certification

Plan

Sécurité d'IP

Protocoles pour sécuriser IP :

- **IPSEC**
- **SSH**



IPSEC

- assure services authentification, confidentialité et intégrité
- défini pour IPv6
- adapté pour IPv4
- plusieurs sous-protocoles

IPSEC - Modes de fonctionnement

- Mode transport
  - protège les échanges entre 2 machines (point à point)
  - traverse NAT
- Mode tunnel
  - encapsule les paquets chiffrés dans de nouveaux en-têtes
  - IPv4/IPv6
  - concu pour VPN
  - masque adresses IP réelles
  - protection contre le rejeu
- IPSEC intégré dans noyaux Linux récents (> 2.6)

Associations de sécurité (SA)

- Conserve les paramètres de sécurité d'une connexion
  - authentification des pairs
  - chiffres employés
  - clés . . .
- **ISAKMP**
  - Internet security association and key management protocol
  - établit, négocie, modifie ou supprime des SA
  - utilise IKE
- **IKE**
  - Internet Key Exchange
  - permet d'établir des SA
  - création manuelle des SA possible

IPSEC – Protection données

- **AH**
  - Authentication Header : authentifie les paquets en ajoutant une somme de contrôle (MAC) de l'en-tête IP jusqu'à la fin du paquet
  - intégrité des données transmises
  - assure une protection contre le rejeu
- **ESP**
  - Encapsulating Security Payload chiffre toutes les données de la couche 4 (transport)
  - ESP encapsule les données entre un en-tête et en en-queue
  - en mode tunnel, les données sont un paquet IP
- **IPcomp**
  - IP Payload compression qui compresse un paquet avant son chiffrement par ESP

SSH

- Secure SHell
- telnet sécurisé
- authentification par clé publique/privée (RSA/DSA)
- authentification client par
  - clé publique/privée (défi)
  - mot de passe chiffré
- applications :
  - export X11
  - SFTP/SSHFS
  - transfert de ports (tunnel)
  - sauts
  - franchissement firewalls

Apprentissage TP2

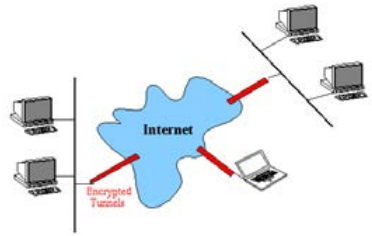
Résumé

- On a vu comment :
  - construire des services de sécurité
  - composer ces services
  - les utiliser pour sécuriser un protocole (donc un service réseau)
- Méthodes pour sécuriser un (ensemble de) serveur(s)
- Comment sécuriser un réseau ?

Plan

VPN

- Réseau privé qui utilise Internet pour connecter :
  - des pairs distants
  - des sites distants
- VPN utilise des connexions virtuelles routées au travers d'Internet vers l'entité distante



Configuration `openvpn` dans TP5

Que fait un VPN ?

- Etend la connectivité géographique
- améliore la sécurité
- réduit les coûts par rapport à un WAN dédié
- simplifie la topologie réseau

Exemple

iface client VPN 192.168.0.4 vers LAN et iface virtuelle de VPN en 10.0.0.4, le serveur est en 10.0.0.1 coté VPN et d' LAN 192.168.1.4. Bridge entre ifaces client et serveur, les réseaux 192.168.0.0/24 et 192.168.1.0/24 se verront. Le client aura accès au réseau 192.168.1.0/24 vec les mêmes filtres que la machine 192.168.1.4 (et recip.)

VPN – modes de fonctionnement

passent par l'interface `tun/tap` ; différence au niveau de la couche OSI

- **Mode routé (routed) :** met en relation des machines distantes. travaille au niveau de la couche 3 (réseau), ie au niveau d'IP. utilise interface `tun`. Etablit route spécifique entre adresses réseau différentes. Pas de broadcast.
- **Mode pont (bridge) :** relie des réseaux distants ; travaille au niveau de la couche 2 (liaison) par protocole dédié `PPTP`, `EoIP`, `IPSec`. utilise interface `tap`. ifaces VPN et LAN liées entre elles en une seule entité ; adresse VPN donnée dynamiquement au client. Routage entre réseaux fait par tables de routage au niveau du serveur VPN. Permet le broadcast et assure transparence complète.

TUN/TAP

TUN/TAP représentent des périphériques réseaux dans un réseau virtuel.

- **TAP** : ou network tap simule un périphérique de lien réseau et agit sur les paquets de la couche 2 (liaison des données), comme les trames ethernet
- **TUN** ; ou network tunnel simule un périphérique de lien réseau et agit sur les paquets de la couche 3 (IP)

l'interface `tap` est utilisée pour réaliser un pont réseau (bridge) et l'interface `tun` faire du routage.

Les paquets envoyés par l'OS au périphérique `tun/tap` sont gérés par un programme de l'espace utilisateur qui peut aussi réaliser l'opération inverse, i.e. injecter les paquets vers l'OS.

Plan

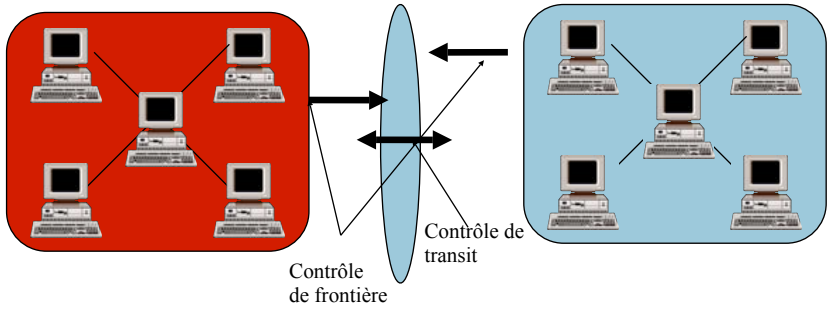
Contrôle des frontières

- Problème principal des LAN connectés à Internet
- **solution** : utiliser des coupe-feux (firewall) avec :
  - filtres de paquets
  - proxy
  - mécanismes cryptographiques
- tout en diminuant le nombre de points d'entrée

Filtres de paquets

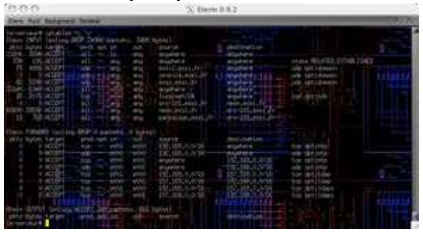
- Fonction assurée par routeurs ou hôtes dédiés.
- Principe du contrôle :
  - redistribuer, effacer et/ou tracer chaque paquet
  - selon sur les informations d'en-tête des paquets
    - adresse source et destination
    - direction (entrée/sortie) par rapport au LAN
    - type d'application par numéro de port
  - en conjonction avec TCP/IP
  - généralement au niveau du noyau de l'OS

Filtrage de paquets



Différents types de pare-feu

Filtre de paquets



Filtrage applicatif

Fonction d'un pare-feu

Un pare-feu désigne un logiciel et/ou un matériel (appliance), qui a pour fonction de faire respecter la politique de sécurité du réseau. Celle-ci définit quels sont les types de communications autorisés ou interdits.

IPtables/netfilter ... nftables

Firewall de paquets disponible sous linux.
Composé de règles de filtrage

- chaîne : liste ordonnée de règles
- chaque règle exprime une condition
- si règle i ne s'applique pas, consulter règle $i + 1$
- une fois épuisé l'ensemble des règles, appliquer la politique par défaut de la chaîne (`ACCEPT`,`DROP`)

Instructions sur les chaînes :

- `-N` créer nouvelle chaîne
- `-X` effacer chaîne vide
- `-P` changer politique
- `-L` afficher règles
- `-F` vider les règles
- `-Z` réinitialiser les compteurs des règles

IPtables – opérations sur les règles

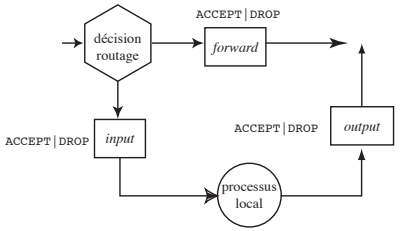
options de manipulations de règles à l'intérieur d'une chaîne :

- `-A` ajouter une nouvelle règle
- `-D` effacer une règle à une certaine position
- `-I` insérer une nouvelle règle à une position donnée
- `-R` remplacer une règle à une certaine position

IPtables – chaînes prédéfinies

`iptables` filtre les paquets qui traversent une machine au moyen des chaînes : `INPUT`, `OUTPUT` et `FORWARD`.

Apprentissage TP1 (et suivants)



IPtables – routage

Un paquet arrivant sur une NIC, le noyau examine sa destination, par le routage.

- Destiné à la machine, il traverse la chaîne `INPUT`. S'il est autorisé à poursuivre son chemin (par un `ACCEPT`), il est traité par le processus local auquel il est destiné. Si la décision est `DROP`, le paquet est supprimé.
- Destiné à une autre interface, le paquet traverse la chaîne `FORWARD` et, s'il est accepté, il poursuit son chemin. Si le `forwarding` n'est pas activé ou si on ne sait pas comment transmettre ce paquet, le paquet est supprimé.
- Un processus local exécuté par la machine peut également envoyer des paquets traités par la chaîne `OUTPUT`.

Utilisation du `ack` bit

- dans `tcp`, le bit d'acquittement accuse réception du datagramme précédent
- dans le datagramme d'ouverture de session, ce bit est à 0
- en bloquant le premier datagramme, toute session est impossible
- en filtrant les paquets `tcp` entrants sans `ack bit`, on interdit les connexions entrantes et on autorise les sortantes
- n'existe pas en `udp`
- appelé `established` en langage routeur ou pour les firewall

Application aux firewalls

```
iptables -A INPUT -p tcp --sport 80 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
```

n'empèche pas le scan par un pirate qui scanne depuis le port 80 (il est ouvert pour autoriser la connexion sur le serveur). Mais cette connexion n'a pas été initiée. Pour bloquer :

```
iptables -A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
```

La seconde règle autorise d'initier la connexion au serveur.

Nouveauté `ufw`

Depuis 2015, un nouveau firewall est apparu dans le monde `debian` : `Uncomplicated Firewall` qui est une surcouche à `iptables` ou `nftables`, gérée par la commande `ufw`. Le principe général est comparable à `iptables`.
Une politique par défaut est définie

```
# ufw default deny incoming
# ufw default allow outgoing
```

On autorise l'accès à un service hébergé sur la machine (comme `ssh` qui écoute sur le port 22), par la directive `allow`

```
# ufw allow ssh
```

En utilisant la directive `deny`, on refuserait la connexion à un service avec la même syntaxe.

Mise au point `ufw`

On accède à la liste numérotée des règles actives par :

```
# ufw status numbered
```

qui affiche

```
Status: active
     To                         Action      From
     --                         ------      ----
[ 1] 22                         ALLOW IN    Anywhere
```

Il est possible de changer la verbosité par :

```
# ufw status verbose
```

Une règle inutile ou obsolète pourra être supprimée par la directive `delete` en précisant le numéro :

```
# ufw delete 1
```

Et sous BSD ?

3 mécanismes différents :

- **IPFILTER** : commande `ipf`
- **IPFIREWALL** : commande `ipfw`
- **PacketFilter** : commande `pf`

Il faut choisir lequel utiliser dans `/etc/rc.conf`, pour `ipfw` :

```
firewall_enable="YES"
```

et sa politique par défaut (`firewall_type="open"`) parmi :

- `open` : laisse passer tout le trafic
- `client` : ne protège que l'hôte
- `simple` : protège le réseau
- `closed` : empèche tout le trafic sauf `lo0` (défaut)
- `filename` : chemin absolu du fichier de règles

IPFIREWALL

- Composé de règles de filtrage ordonnées
- chaque règle exprime une condition
- si la règle *i* ne s'applique pas, consulter la règle $i + 1$ (numérotation manuelle)
- appliquer la dernière règle qui décrit la politique par défaut

Instructions sur les règles :

- `add` : ajoute règle
- `delete` : efface règle
- `list` : affiche règles
- `flush` : vide les règles
- `zero` : réinitialise compteurs

Fonctionnement

```
# ipfw l
        00100 allow ip from any to any via lo0
        00200 deny ip from any to 127.0.0.0/8
        00300 deny ip from 127.0.0.0/8 to any
        65000 allow ip from any to any
        65535 deny ip from any to any
```

On reçoit : `IP socrate.unice.fr.17500 > neon.unice.fr.23:` telnet sur `neon (hôte local)` arrive sur `en0`. On cherche la première règle qui s'applique (`65000`) et on laisse donc passer le paquet.
Même procédé pour la réponse avec la même règle.

Ecrire les règles

`ipfw add 65 allow tcp from 205.238.129.221 23 to 1.2.3.4 out via en0`

Dans `/etc/firewall.local` on omet le `ipfw`. On peut mettre le nom `dns` à la place de l'IP, changer bien des choses. Par exemple `me` représente la machine, màj dynamique si `dhcp`.

```
# Allow bruno.martin.net to talk to me and conversely
add 100 allow ip from bruno.martin.net to any in via en0
add 105 allow ip from any to bruno.martin.net out via en0
# Allow this machine to make DNS queries
add 5000 allow udp from me to any 53 out via en0
add 5005 allow udp from any 50 to me in via en0
# Allow anyone to ssh to me and allow me to respond
add 10000 allow tcp from any to me 22 via en0
add 15673 allow tcp from me 22 to any via en0
# Allow and log everything else to look for
add 20000 allow log ip from any to any
```

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/firewalls-ipfw.html

Et sous BSD ?

3 mécanismes différents :

- **IPFILTER** : commande `ipf`
- **IPFIREWALL** : commande `ipfw`
- **PacketFilter** : commande `pf`

Il faut choisir lequel utiliser dans `/etc/rc.conf`, pour `pf` :

```
pf_enable="YES"          # Enable PF
pf_rules="/etc/pf.conf" # rules definition file for PF
```

`pf` permet de définir des macros (préfixées par `$`) qui simplifient l'écriture de règles comparables. Pour plus de détails, voir http://srobb.net/pf.html.

Exemple de fichier `pf`

```
block in all
pass out all keep state
tcp_services_out = "{ssh, smtp, domain, www, auth, imaps}"
tcp_services_in = "{ ssh }"
udp_services = "{ domain }"

set skip on lo0
set skip on lo1
#pass out proto tcp to port $tcp_services_out keep state
#pass proto udp to port $udp_services
pass in proto tcp to any port $tcp_services_in keep state
```

Limites d'un pare-feu

- toutes les communications doivent passer par le pare-feu
- le pare-feu doit être convenablement configuré
- éviter le contournement (modem, gsm...)
- éviter l'utilisation de clés usb, ordinateurs portables
- tenir un journal (logs)
- détecter les anomalies et/ou les intrusions

Plan

Détection d'intrusion

Un système de détection d'intrusion (ou IDS : Intrusion Detection System) est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte). Il permet ainsi d'avoir une connaissance sur les tentatives réussies comme échouées des intrusions. Classés en plusieurs catégories :

- NIDS : Network based IDS qui surveillent l'activité au niveau du réseau (exemple : `snort`)
- HIDS : Host based IDS qui surveillent l'activité au niveau des hôtes
- IDS hybrides : qui combinent les 2 précédents

NIDS

Snort is an open source network IDS, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plugin architecture. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's smbclient. Snort has three primary uses. It can be used as a straight packet sniffer like tcpdump(1), a packet logger (useful for network traffic debugging, etc), or as a full blown network intrusion detection system.

Plan

Résumé

- On a vu comment :
  - construire des services de sécurité
  - composer ces services
  - les utiliser pour sécuriser un protocole (donc un service réseau)
  - assurer un filtrage
    - au niveau applicatif
    - au niveau de la couche réseau
- La question devient : Pour faire quoi ?

Politique de sécurité

But : informer les utilisateurs, personnels et responsables, des conditions à satisfaire pour protéger les avantages technologiques et en information de l'entreprise.

Définit les mécanismes de protection et sert de fil conducteur pour configurer et auditer les SI.

Elle commence généralement par la phrase :

Tout ce qui n'est pas autorisé est interdit

Caractéristiques d'une politique de sécurité

1. doit être implémentable par l'administrateur
2. doit être améliorable (mesures de sécurité/sanctions)
3. doit définir les domaines de responsabilité de chacun

Classification des risques

1. **Nul :** risque jugé non significatif
2. **Faible :** événement génèrant une nuisance organisationnelle, des pertes financières faibles, peu gênant pour l'utilisateur
3. **Sensible :** événement occasionnant des pertes financières significatives, nuisible à l'image, gênante pour l'utilisateur
4. **Critique :** événement occasionnant des pertes financières innacceptables, une perte de clientèle
5. **Stratégique :** événement susceptible d'entraîner un arrêt immédiat d'une activité de l'entreprise

Evaluation des risques

Une équation simple :

$$\text{Risque} = \text{Menace} \times \text{Vulnérablité} \; [\times \text{Coût}]$$

- **Menace** : ce contre quoi on veut se défendre (DoS,...)
- **Vulnérabilité :** faiblesse connue de l'architecture de sécurité (trop de points d'accès, faible authentification,...)
- **Coût :** impact financier

Voir également là.

Top 5 des menaces en 2020

- DNS hijacking ($\rightarrow$ MiTM)
- Rançongiciels
- Remote Access Trojan
- Office 365 Phishing
- Digital Extorsion Scams

Hi, your account has been infected! Renew the pswd right this moment!
You do not heard about me and you obviously are certainly wondering for what reason you are getting this email, is it right?
I'm ahacker who openedyour emailand devicesnot so long ago.
Don't attempt to contact me or try to find me, it is impossible, since I directed you an email from YOUR own hacked account.
I've created virus to the adult videos (porno) site and guess that you have watched this site to have fun (think you understand what I really mean).
During the time you have been keeping an eye on vids, your browser started out operating like a RDP (Remote Control) with a keylogger which provided me the ability to access your display and webcam.
Consequently, my softobtainedall information.
You wrote passcodes on the web services you visited, I already caught them.
Surely, you could possibly modify each of them, or have already changed them.
However it doesn't matter, my program updates needed data every 5 minutes.
And what did I do?
I generated a backup of every your device. Of all the files and personal contacts.
I formed a dual-screen record. The 1st part displays the film you had been watching (you have got the perfect preferences, huh...), the 2nd screen demonstrates the video from your own web camera.
What do you have to do?
Great, in my opinion, 1000 USD is a fair price for our very little riddle. You'll make the payment by bitcoins (if you do not recognize this, search "how to purchase bitcoin" in Google).
My bitcoin wallet address:

13cas4mnDPoNBDS3YJsthyfpfmEShDxMSD

(It is cAsE sensitive, so copy and paste it).
Attention:
You will have only 48 hours to make the payment. (I have an unique pixel in this e-mail, and at this time I understand that you've read this email).
To tracethe reading of a letterand the actionsin it, I installeda Facebook pixel. Thanks to them.
(Everything thatis usedfor the authorities may also helpus.)
In case I do not get bitcoins, I shall undoubtedly send your video to each of your contacts, along with relatives, co-workers, and so forth?

Tant en interne qu'en externe (voir Cisco Threat Report)

Mise en œuvre d'une politique de sécurité

1. identifier les besoins en terme de sécurité, les risques et les conséquences
2. trouver les règles et procédures à mettre en œuvre pour les risques identifiés
3. surveiller et détecter les vulnérabilités du SI et effectuer une veille technique
4. définir les actions à entreprendre et qui contacter en cas de détection d'une menace.

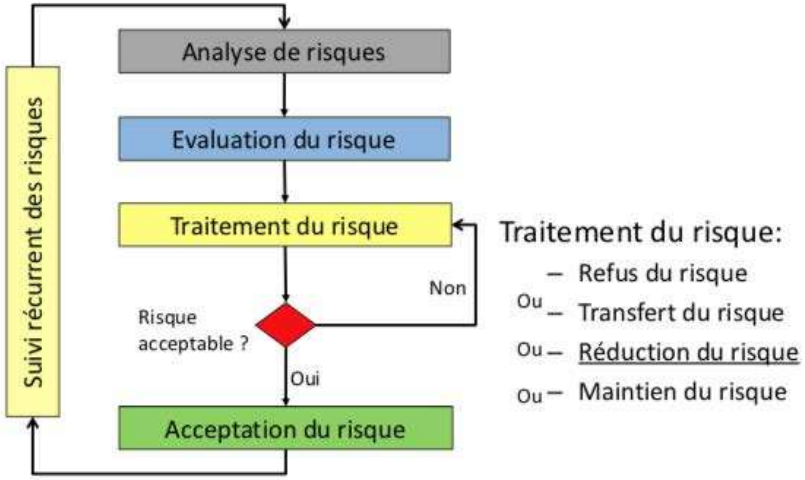
Gestion des risques

Consiste en la réalisation et le maintien à jour :

- de l'**inventaire** des **actifs**
- de l'expression des **besoins de sécurité** de ces actifs
- de l'analyse des risques pesant sur les actifs
- du traitement de ces risques pour les réduire

traite les risques par des méthodes MEHARI ou EBIOS
informe et sensibilise les personnels (chartes, lettre de sécurité, RSS RSSI...)

Processus de gestion des risques



Actifs

Regroupent les biens de l'organisme et ses ressources humaines ; 3 types :

- actifs gérés au travers du SI (infos et processus métier)
- actifs techniques constituant le SI (logiciels, matériels, moyens de comm)
- actifs relatifs à l'environnement (personnes et bâtiments)

Le propriétaire d'un actif = personne ou entité ayant accepté le contrôle de la production, mise au point maintenance, utilisation et protection des actifs

Actifs/Assets ?

Actifs généralement inventoriés dans les entreprises :

- 96% actifs physiques (matériel info/comm)
- 93% logiciels
- 82% informations
- 57% services info/comm
- 41% personnels et leurs compétences
- 20% valeurs immatérielles (réputation, image)

Inventaire des actifs

- coût d'achat de l'actif
- coût de remplacement
- valeur de la propriété intellectuelle
- coût de maintenance
- coût des responsabilités si des données personnelles sont compromises

Qui définit une politique de sécurité ?

Tous les membres d'une même organisation doivent adhérer à la politique de sécurité pour que celle-ci devienne effective.
Plus spécifiquement :

- l'administrateur de sécurité du site
- le personnel technique
- les chefs de service
- le groupe d'audit de sécurité
- des représentants des utilisateurs
- le directeur général
- un conseiller juridique le cas échéant

Contenu d'une politique de sécurité

- politique d'achat de matériel de sécurité
- une politique de respect des droits individuels (lecture d'e-mails)
- définir politique d'accès et droits sur les données avec des messages d'alerte adéquats
- une politique de gestion des comptes qui définit les responsabilités et les mesures d'audit
- définir une politique d'authentification des utilisateurs
- définir la disponibilité des ressources pour gérer les pannes et les mises à jour logicielles et matérielles
- définir une charte de maintenance du système et des ressources
- tenir à jour un cahier des intrusions et de leur type

Flexibilité d'une politique de sécurité

Il faut assurer la viabilité de la politique de sécurité. Celle-ci doit être basée sur un concept d'architecture de la sécurité. Elle doit être la plus indépendante possible de matériels et de logiciels spécifiques qui peuvent être facilement remplacés.

Penser qu'il y a des exceptions à chaque règle. Il faut essayer de tenir à jour une liste des exceptions aux règles de sécurité. P.e. dans quel type de situation un administrateur a le droit d'explorer le contenu d'un compte utilisateur.

Exemples de politique de sécurité

- La porte de votre appartement
- Le courrier postal
- Accès université
- Accès INRIA

La politique de sécurité est formalisée par des modèles de sécurité. C'est expression formelle (mathématique) de la politique de sécurité. (exemple des droits d'accès sous UNIX) ou modélisés par des langages comme ORBAC.

Exemple « léger »

- Matériel, périphériques et équipements
  - utiliser un onduleur
  - supprimer les données des vieux équipements et contrôler l'infrastructure réseau
  - verrouiller chaque poste de travail
- travail à distance
  - définir le cadre de travail d'un collaborateur extérieur
  - sensibiliser le personnel aux risques de l'utilisation d'un ordinateur portable et du travail à distance
- contrôle de l'accès au SI et à ses contenus
  - avoir une authentification uniforme et centralisée
  - classifier l'information ; l'associer à des profils d'utilisateurs
  - bien définir les rôles des utilisateurs
  - avoir une politique de sélection des mots de passe
  - placer les serveurs et équipements réseau dans des locaux à accès restreint

Exemple « léger »

- traitement de l'information
  - faire installer et gérer le réseau par des personnels qualifiés
  - limiter les actions d'administration à du personnel qualifié
- email et accès Internet/Intranet/Extranet
  - utiliser des détecteurs de virus
  - utiliser des outils de confidentialité
  - mettre en place un firewall
  - traiter avec précaution tout mail non sollicité
  - vérifier `from` et `to` de tout email
  - limiter la taille d'expédition des messages

Vers une normalisation (ISO 27001-2)

ISO 17999 en 2000 maintenant 27002 pour la sécurité des SI. Destinée aux dirigeants, aux directeurs de système d'information et aux responsables sécurité (Chief Security Officer, RSSI). Code de bonnes pratiques pour la gestion de la sécurité de l'information.
ISO 27001 : norme de gestion de la sécurité de l'information : Technologies de l'information- techniques de sécurité - Systèmes de gestion de sécurité de l'information - Exigences.
Tout comme la norme ISO9000 pour la qualité, la norme ISO17999 a pour objectif d'établir un label de confiance reconnu de tous en ce qui concerne la sécurisation de l'information sous un aspect global.

Vers une normalisation

ISO 17999 : importance particulière à des aspects de la sécurité :

- le support des dirigeants quant à la mise en œuvre d'une politique de sécurité et la détermination des moyens humains
- l'identification des menaces propres à l'organisation et l'évaluation des risques associés
- la classification des informations afin de ne déployer les moyens que sur celles qui le nécessitent
- les dispositions prendre pour instaurer une "culture sécurité".

En conjonction avec des guides techniques :

- ISO13335 : concepts et modèles pour la gestion de la sécurité
- ISO14516 : gestion et utilisation des services de certification
- ISO15408 : critères d'évaluation de la sécurité
- ISO18044 : gestion des incidents de sécurité

ISO 27002

Découpée en 15 articles (chapitres) ; 200 CHF ; aux US : NIST handbook : Introduction to computer security

- 4 qui définissent le cadre de la norme
- 11 articles qui proposent 133 mesures définissant les objectifs de sécurité et les mesures à prendre :

- politique de sécurité
- organisation de la SI
- gestion des biens
- sécurité et RH
- gestion télécom
- contrôle accès
- acquisition, dév. maint. SI
- gestion des incidents
- continuité de l'activité
- conformité

ISO 27002 – critères de succès

- pointe et évalue les risques encourus
- mise en œuvre compatible avec culture entreprise
- soutien et engagement visible de la Dir.
- compétence et moyens pour mettre en place une politique de sécurité
- formation appropriée à tous les échelons de l'entreprise
- accès pour tous aux normes et directives de sécurité

Plan

Termes

- **Penetration test :** méthode d'évaluation de la sécurité d'un hôte ou réseau en simulant une attaque. Pour ça :
  - rechercher les points d'accès
  - rechercher les vulnérabilités

  selon différentes approches :
  - **Black box (covert)** : pas de connaissance de l'infrastructure ; effacer ses traces
  - **White box (overt)** : infrastructure connue, avec RSSI
  - et les variantes entre les deux (grey box).
- **Forensic** : Terme adapté de l'anglais «computer forensics», l'expression « investigation numérique » représente l'utilisation de techniques spécialisées dans la collecte, l'identification, la description, la sécurisation, l'extraction, l'authentification, l'analyse, l'interprétation et l'explication de l'information numérique.

Phases du PenTest

- Pre-engagement interaction : négociation avec client : "contrat"
- Intelligence gathering : récupération de toutes les infos possibles sur le client (réseaux sociaux, scan, footprint,...)
- Threat modeling : utiliser les infos de l'IG pour identifier les vulnérabilités, choisir les attaques en fonction des buts cherchés
- Vulnerability analysis : trouver les attaques possibles en fonction de l'analyse des ports et des vulnérabilités,...
- Exploitation : réalisation d'exploits
- Post exploitation : attaques en whitehat
- Reporting : rapporter le détail des opérations menées

Intelligence gathering

- un bon hacker (bidouilleur) programme un outil pour scanner (explorer) le réseau
- il le publie sur Internet
- un script kiddie (novice) l'utilise pour trouver des systèmes vulnérables ou des points d'accès

Scanner de ports : `nmap`

Un port scanner peut parcourir une grande plage d'adresses IP et retourner les ports ouverts (donc les services accessibles) ainsi que les version d'OS

`nmap`, modes de fonctionnement

vanilla tentative de connexion sur tous les ports

strobe cible certains ports spécifiques

fragment packets limitation à des paquets fragmentés (pour traverser certains fw)

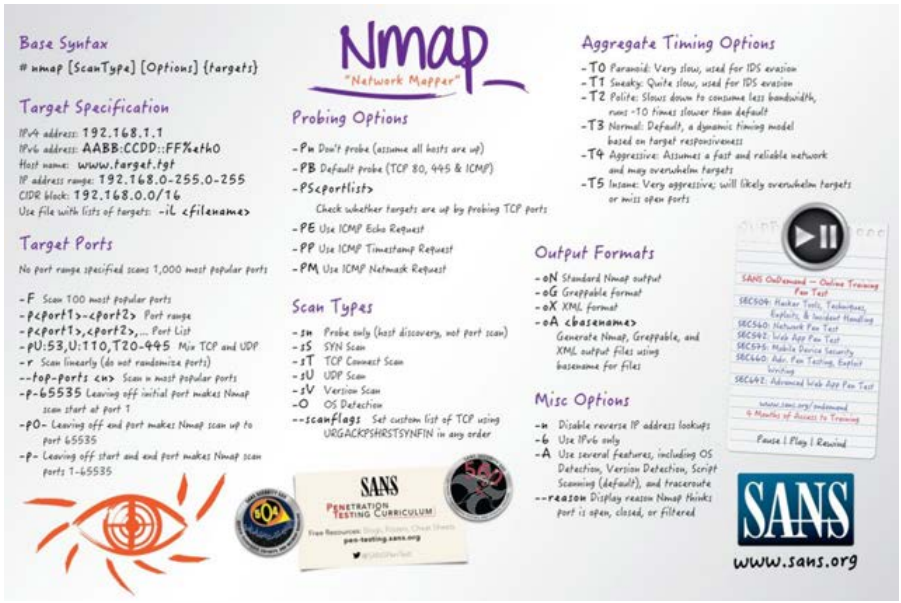
udp recherche des ports `udp`

sweep connexion sur le même port d'un ou plusieurs PC

FTP bounce imite le fonctionnement d'un serveur `ftp` pour paraître légitime

stealth permet d'augmenter le discretion en empêchant partiellement le fonctionnement des mécanismes de log

Scanner de ports : `nmap`



OS detection

Tehniques standard : par banner grabbing ; sinon

- connexion `smtp`, `snmp` ou `telnet` pour examiner les réponses du serveur
- dans `nmap` fingerprint de la pile `tcp/ip` qui permet d'identifier la réponse du système à des paquets `tcp` avec des drapeaux particuliers

Vulnérabilités

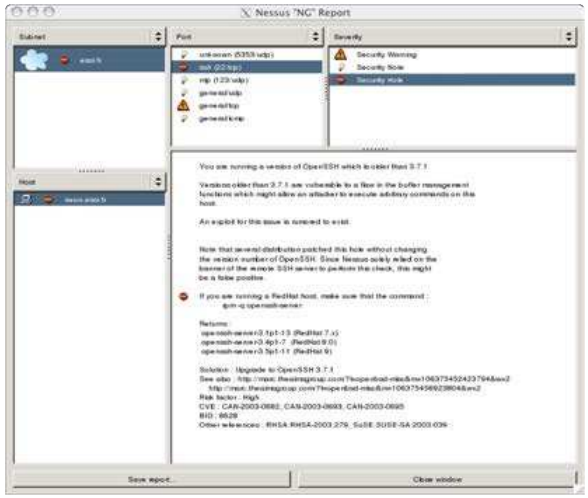
- le novice utilise ensuite une liste d'IP vulnérables pour accéder au système
- selon les faiblesses, il peut éventuellement créer/utiliser un compte ou un accès privilégié
- il l'utilise pour acquérir de nouveaux privilèges et pour pirater de nouveaux systèmes connectés à sa 1re victime
- exemple : se faire passer pour une machine du réseau attaqué (avec `wireshark` ou `ettercap`)

Scanner de vulnérabilités – Nessus

The "Nessus" Project aims to provide to the internet community a free, powerful, up-to-date and easy to use remote security scanner, software which will audit remotely a given network and determine whether someone may break into it, or misuse it. Nessus does not take anything for granted. That is, it will not consider that a given service is running on a fixed port - a web server on port 1234, will be detected it and its security tested. Nessus is fast, reliable and has a modular architecture that allows you to fit it to your needs. Nessus works on Unix-like systems (MacOS X, FreeBSD, Linux, Solaris and more) and a Windows version called NeWT is available.

Nessus : cibler

Nessus : scanner



Nessus - OpenVAS

Nessus est devenu payant (cher) en 2005.
OpenVAS représente la branche « libre » de nessus. Contenait en 2011 plus de 23 000 tests de vulnérabilité, reliés à la base “Common Vulnerabilities and Exposures” CVE qu’on peut interroger
Il est possible d’ajouter des plugins dans le langage NASL, comme dans nessus.
http://www.openvas.org/

Test de vulnérabilité

Connaitre les vulnérabilités permet de déterminer la surface d'attaque

Attaquer

- généralement au moyen de `rootkits`
- un `rootkit` est un terme qui décrit un ensemble de scripts et d'exécutables qui permettent à un pirate de cacher ses agissements et d'obtenir un accès privilégié au système :
  - modifie les `logs`
  - modifie les outils système pour rendre la détection du piratage difficile
  - crée une trappe d'accès cachée
  - utilise le système comme point d'entrée sur les autres hôtes du LAN

Framework Metasploit

Metasploit (écrit en `ruby`) est un framework qui permet :

- de collecter le résultat des différents scanners (port, vulnérabilité,. . .)
- d'automatiser (et de rejouer) des attaques contre des vulnérabilités identifiées (et d'en ajouter)

outil très puissant mais compliqué à utiliser (gui : armitage)
Une petite video pour l'utilisation se trouve à
http://www.youtube.com/watch?v=AG_MeOsnQwM

Les 10 règles de la sécurité

- Sécuriser les points faibles
- Opérer en profondeur
- Bien gérer les cas d'erreur
- Principe du strict minimum
- Cloisonner
- Rester simple
- Encourager le secret
- Il est difficile de garder un secret
- Rester méfiant
- Utiliser les ressources de la communauté

Plan

Qu'est-ce que la virtualisation ?

Ensemble de techniques qui permettent l'exécution de plusieurs systèmes sur une même machine physique.

La couche de virtualisation, l'**hyperviseur**, masque les ressources physiques du matériel pour proposer à un SE des ressources différentes de ce qu'elles sont en réalité.

Rendre un SE indépendant du matériel sur lequel il est installé.

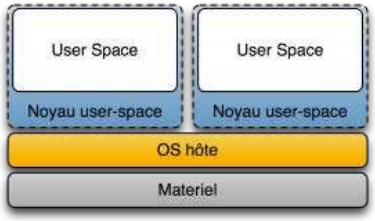
Objectifs de la virtualisation

- réduction coûts et du nombre d'équipements
- réduction du délai de mise à disposition de serveurs
- simplifier administration et gestion
- améliorer le niveau de service et la disponibilité
- créer des environnements de test et de production
- utiliser des SE anciens ou exotiques
- mise en place d'un plan de reprise

Techniques de virtualisation de SE

- noyau en espace utilisateur
- machine virtuelle
- para virtualisation ou hyperviseur
- (matériel : le support de virtualisation est intégré au processeur ou assisté par ce dernier. Surtout pour des mainframes)

Noyau en espace utilisateur

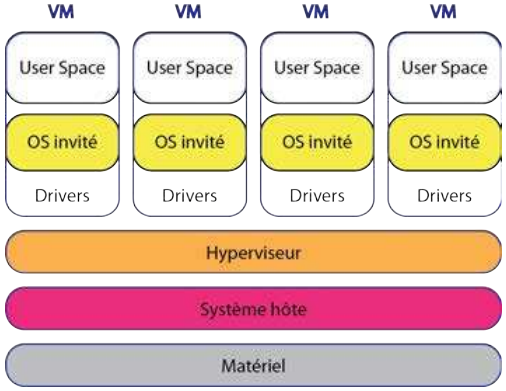


Un noyau tourne comme une application en espace utilisateur ; il a son propre espace mémoire donc le contrôle d'applications. Le SE invité n'est pas indépendant de son hôte et les 2 noyaux sont empilés dans le même système physique. Cas de UML.

Machine virtuelle

Un logiciel émule le matériel et permet de lancer plusieurs OS invités sur le système hôte physique. La mémoire et le CPU sont directement accessibles aux VM. On parle d'hyperviseur de type 2.
Technique qui fait cohabiter plusieurs OS isolés qui communiquent par un réseau émulé.

- KVM
- QEMU
- **VMware**
- VirtualBox
- VirtualPC
- Bochs



Machine virtuelle : mise en réseau

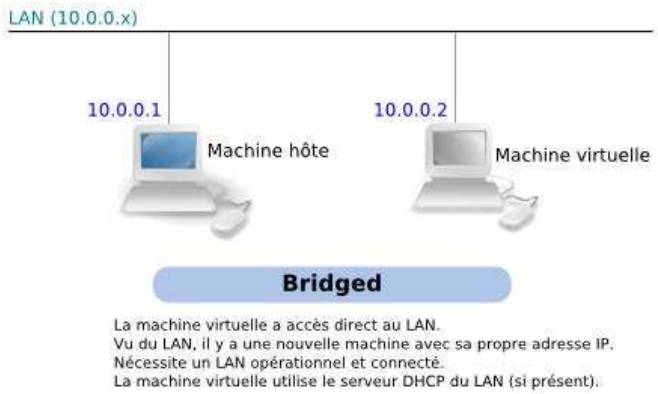
3 modes de fonctionnement principaux

- **NAT** : fonctionnement le plus simple. La VM se comporte comme un hôte connecté à un routeur qui acquiert son IP par `dhcp`. Le logiciel de virtualisation sert de routeur.
- **bridge** : partage l'interface physique de la machine hôte. La VM acquiert son IP par un serveur externe à l'hôte.
- **host only :** la VM ne peut communiquer qu'avec d'autres VM hébergées sur le même hôte, comme si elles étaient connectées par un switch.

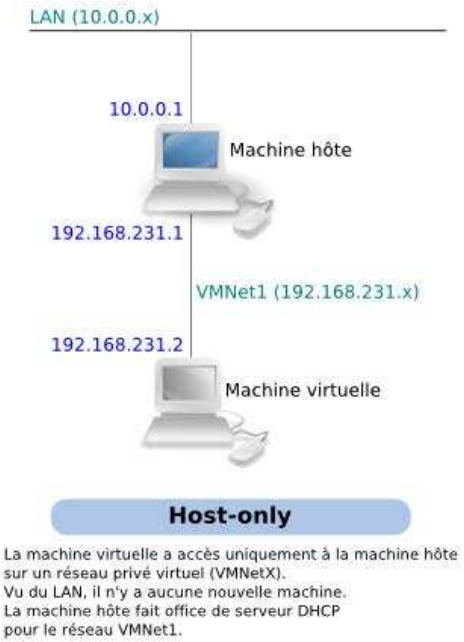
	accès au LAN	adr. IP de LAN
host only	NON	NON
NAT	OUI	NON
Bridge	OUI	OUI

NAT

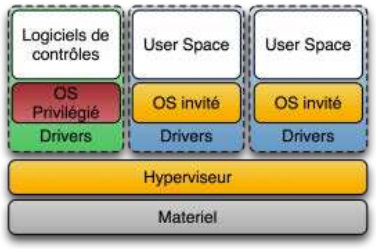
Bridge



Host-only



Hyperviseur (type 1)



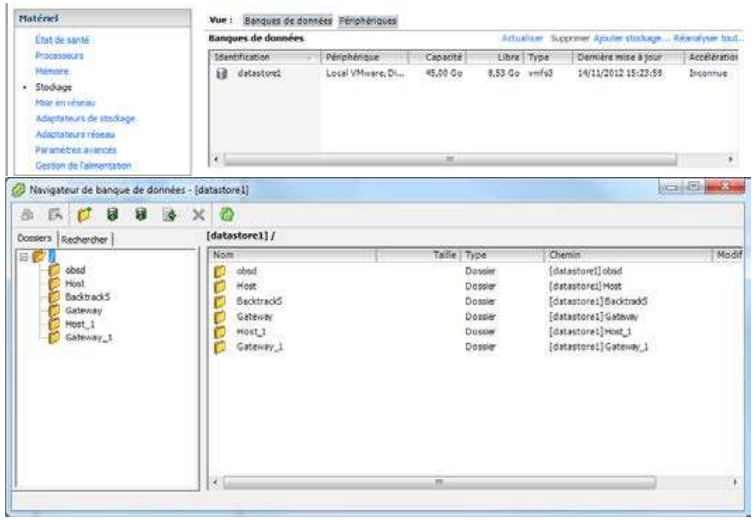
L'hyperviseur est un noyau système léger et optimisé pour la gestion de noyaux des SE invités. On parle aussi d'hyperviseur de type 1 ou natif ou baremetal.

- **VMware** ESX, ESXi
- Xen qui est utilisé en particulier dans QubesOS (voir la présentation)

VMware vSphere

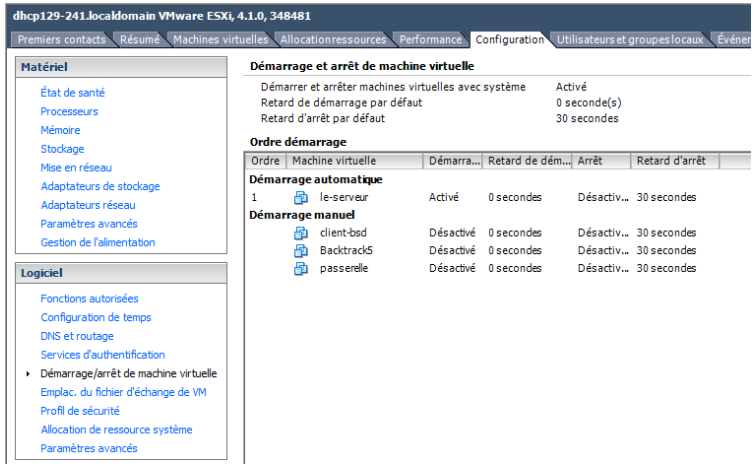
- Serveur hôte ESX/ESXi :
  - cœur de l'infrastructure
  - SE qui permet de faire tourner des VM simultanément
- VCenter server : administration centralisée des ESX
  - supervise et d'administre toutes les activités des serveurs hôtes ESX et des VM
  - indispensable pour les fonctionnalités type VMotion,. . .
- Datastore : espace de stockage uniforme des VM
- vSphere client : GUI utilisateur vers infrastructure :
  - créer, administrer, monitorer les VM et hôtes ESX par connexion directe sur un ESX ou un VCenter
- VSphere Webaccess : GUI web d'admin :
  - réalisation des actions de base d'admin + config des VM

Le Datastore



vSphere

vSphere



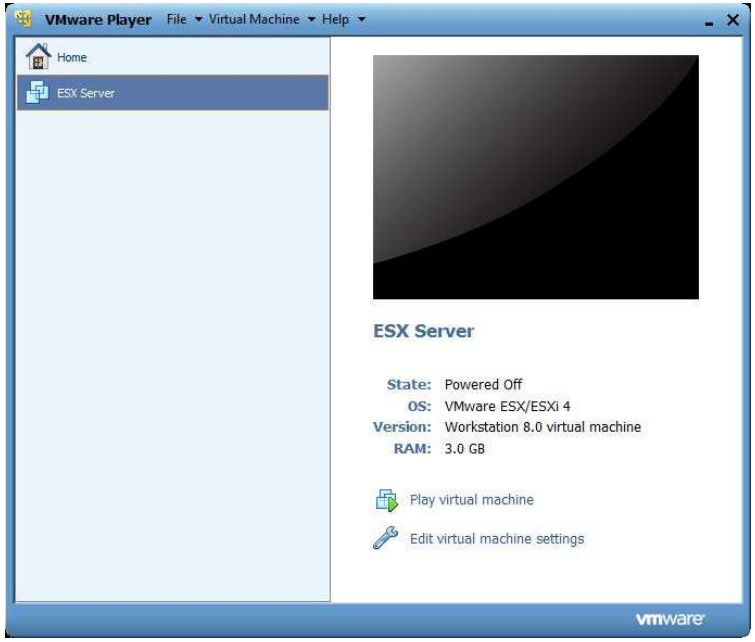
VMware ESX/ESXi

Au cœur de l'infrastructure.
Hyperviseur de type 1 « baremetal » installé directement au dessus du matériel. Pas besoin de SE hôte.
ESX/ESXi est composé :

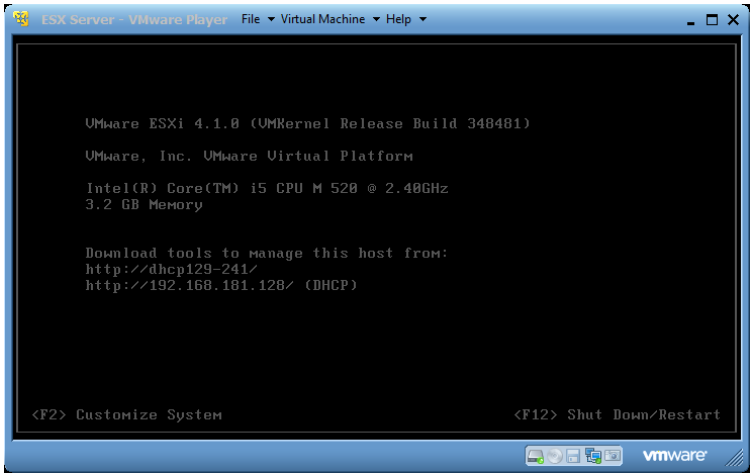
- de la couche de virtualisation
- (d'un service console, pour ESX seulement)
- des VM

Dans notre configuration, on utilise ESXi virtualisé dans une VM qui émule un hardware. ESXi tourne sur un VMplayer qui accède à Internet par NAT.

ESXi sous VMPlayer



ESXi sous VMPlayer



Différences entre ESX/ESXi

- ESX : noyau de virtualisation (vmkernel) enrichi d'une partition de gestion, le système d'exploitation en console (COS). Le COS fournit une interface de gestion intégrée à l'hôte. Cela permet d'assurer des fonctions de surveillance du matériel ou la gestion du système. Taille env. 2Go
- ESXi conserve le vmkernel mais le COS est supprimé. Les service d'infrastructure sont disponible nativement par des modules du vmkernel. Taille env. 150Mo

Rôle du VMkernel

- Moteur de la virtualisation
- Entièrement développé par VMware en 64 bits
- contrôle et gère les ressources matérielles du serveur
- Alloue dynamiquement aux VM le CPU, la mémoire, les accès disques et réseau
- Contient les pilotes de périphériques des composants du serveur physique : NIC, contrôleur de disque, SGF,. . .

Couche de virtualisation VMmonitor

- Contient l'exécution de toutes les instructions du virtual CPU
- Met en correspondance la mémoire VM avec celle de l'hôte
- Intercepte les E/S en provenance des VM et les soumet au VMkernel
- Gère les ressources minimales garanties au démarrage (RAM et HD) et leur isolement

Machines virtuelles sous ESX/ESXi

- Composées d'un SE hôte avec un Virtual Hardware
- Caractéristiques :
  - encapsulation : VM encapsulée dans des fichiers représentant l'ensemble du serveur physique (matériel, virtuel et SE hôte)
  - indépendance matérielle
  - abstraction des composants matériels du serveur physique
  - le virtual hardware configuré dans le SE hôte est identique quelque soit le matériel physique

Une Ubuntu virtualisée

Fichiers d'une machines virtuelles

- Stockage :
  - `.vmdk` : décrit les paramètres physiques du disque virtuel
  - `-flat.vmdk` : correspond au disque virtuel avec son contenu (SE, applications, données)
  - `.rdm` : accès direct au Logical Unit Number
- Mémoire :
  - `.wswp` : swap de la mémoire de la VM
  - `.vmss` : snapshot temporaire de la mémoire vive de la VM
- Configuration :
  - `.vmx` : infos de config matérielle (taille HD, RAM, NICs)
- Snapshot :
  - `0000#-delta.vmdk` : fichier de snapshot
  - `Snapshot#.vmsn` : état VM au moment du snapshot
  - `.vmsd` : info et métadonnées des snapshots comprenant le nom du vmdk et vmsn associés

Virtualisation des composants

- **Stockage** : local ou centralisé (NAS ou SAN) géré par ESXi via le Datastore
- **Mémoire** : sur-allocation mémoire possible (8 VM d'1Go sur hôte avec 4Go de RAM) par pagination, swap et ballooning (permet à la machine physique de récupérer la mémoire inutilisée des VM hôtes et de réallouer les ressources).
- **Processeur** : instructions VCPU de la VM interceptées par le VM manager et transmises au VMKernel qui se charge de l'allocation. Environ 3 VM par cœur.
- **Réseau** : basé sur des switch virtuels (vSwitch) opérant au sein du VMkernel. Permettent de relier les composants physiques du réseau aux composants physiques virtuels. vSwitch est similaire à un switch physique pour le routage

Réseau

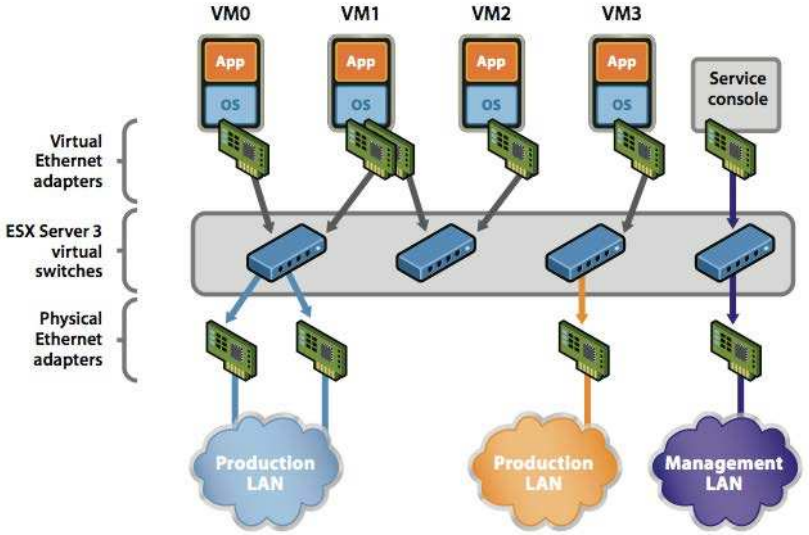
3 types de communications à la création d'un vSwitch :

- **Service Console Port (vSwitch)**
  - utilisé par le COS pour administrer le serveur ESX
  - requiert une IP
  - doit être connecté au moins à une iface physique pour communiquer avec vSphere
- **VMKernel Port (VMK)**
  - Utilisé par vMotion, iSCSI ou NFS
  - requiert une IP
  - doit être connecté au moins à une iface physique
- **Virtual Machine Port Group** :
  - n'a pas besoin d'IP
  - on peut lui connecter de 0 à *n* cartes réseau
  - si aucune carte n'est connectée, le réseau est complètement isolé de l'extérieur. Communications au niveau du VMkernel

Communications réseau des VM

- Chaque VM a une ou plusieurs interfaces VNIC
- Les VNIC ont une adresse MAC, une IP, un driver et suivent ethernet
- Les vSwitchs possèdent des ports virtuels ou des groupes de ports
- Les VM connectent leur VNIC à un port virtuel du groupe des vSwitchs
- Le réseau physique passe par des Uplink ou VMNIC

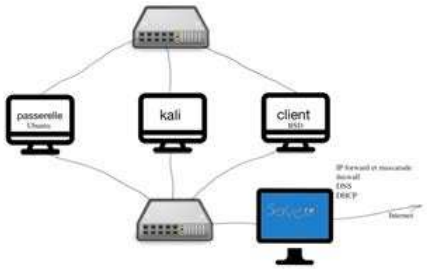
Mise en réseau

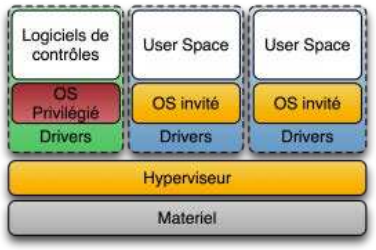


Architecture réseau VMware

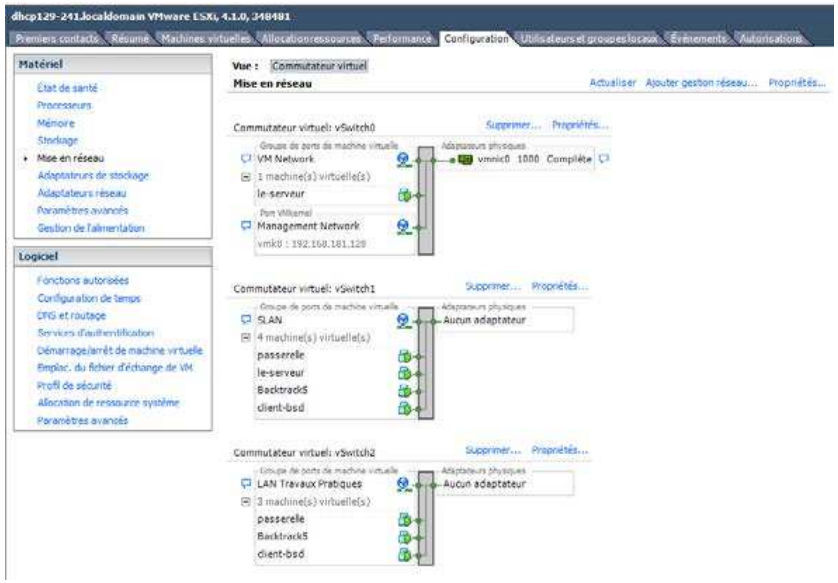
- Le réseau peut être le goulot d'étranglement dans une architecture virtuelle
- multiplier les cartes réseau (physiques)
- créer des vSwitch séparés pour le réseau de gestion et celui des VM (sécurité)

Virtualisons les VM !

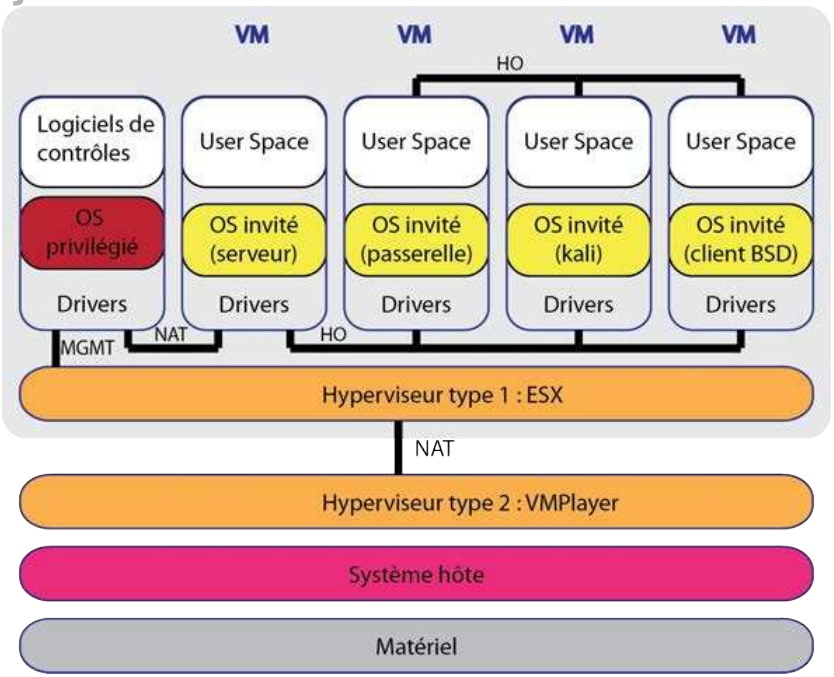




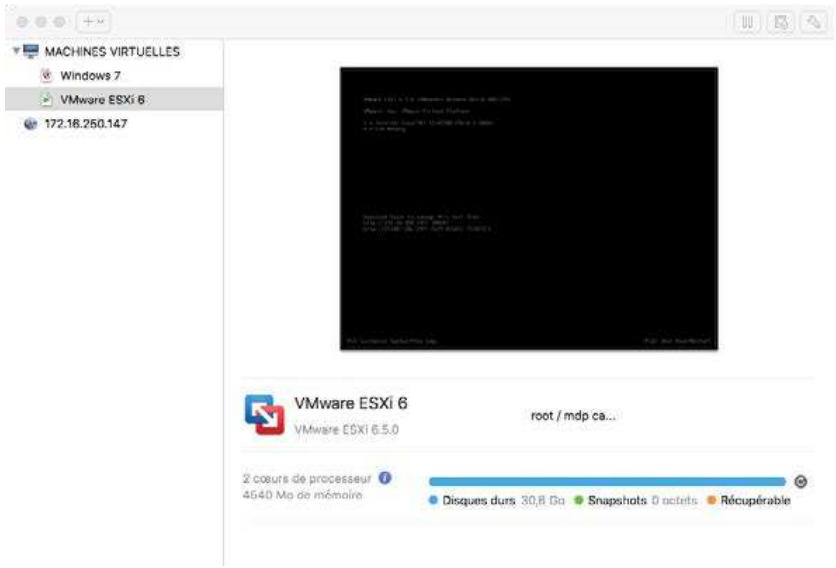
Réseau TP



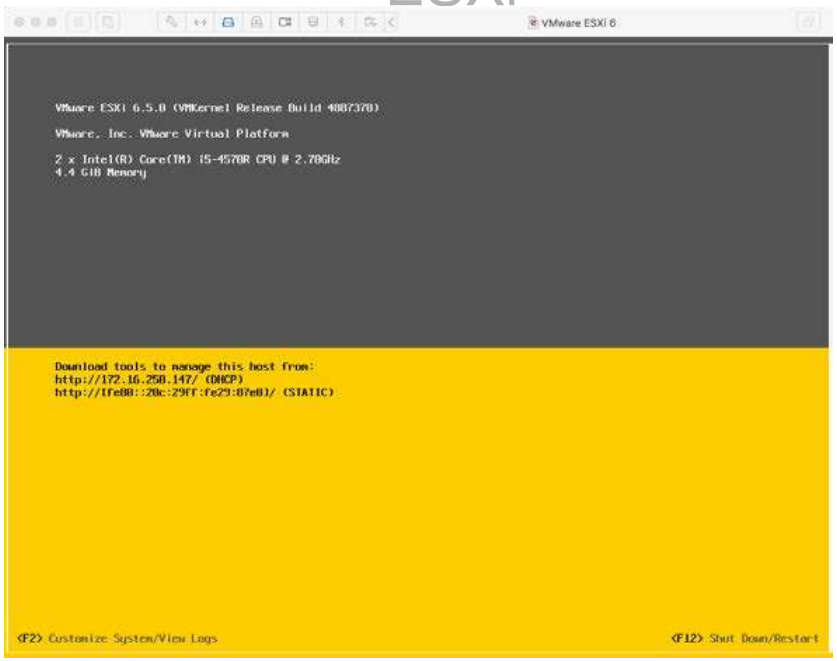
Ajout d'une couche de virtualisation



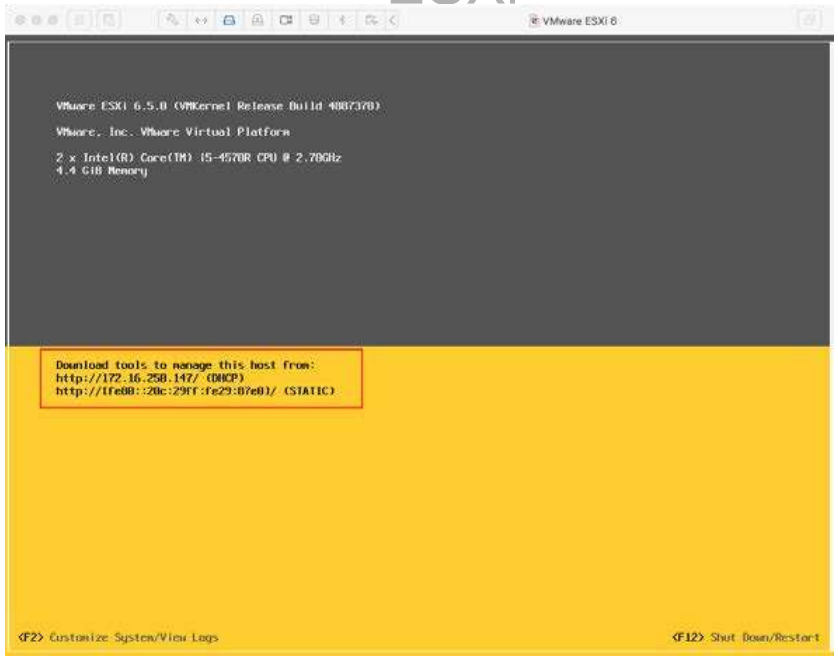
Ajout d'une couche de virtualisation : résultat



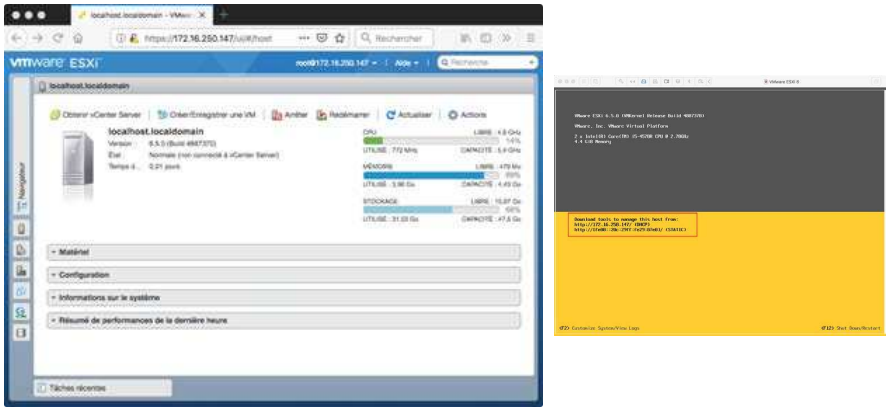
Ajout d'une couche de virtualisation : ESXi



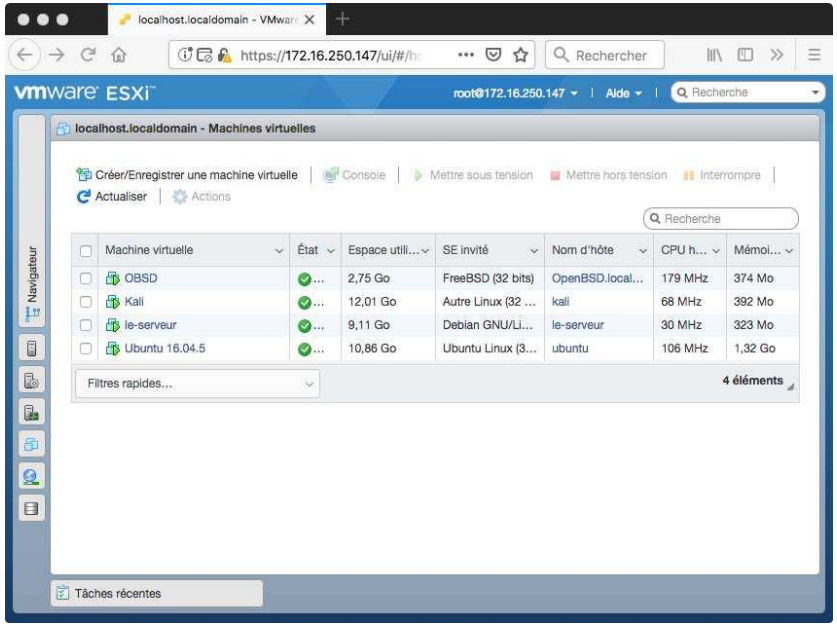
Ajout d'une couche de virtualisation : ESXi



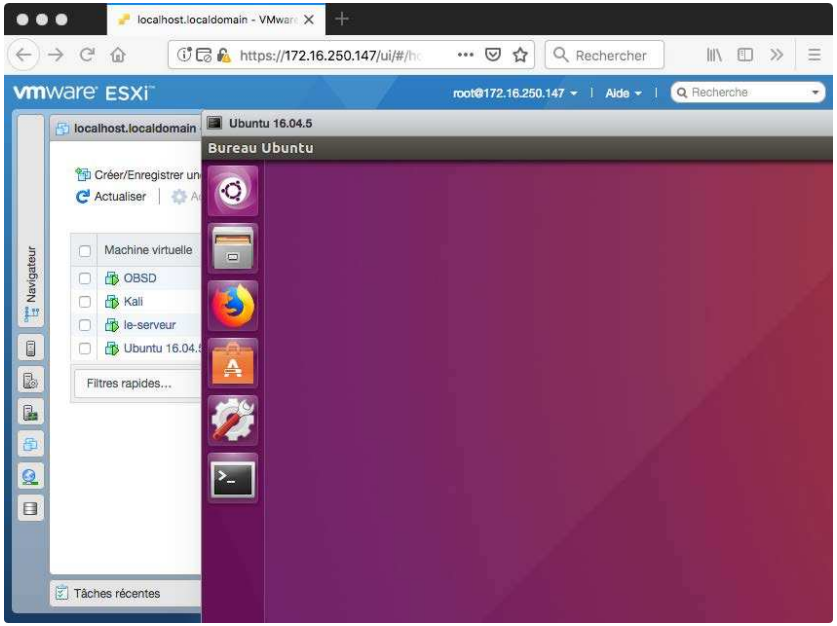
Connexion à l'ESXi



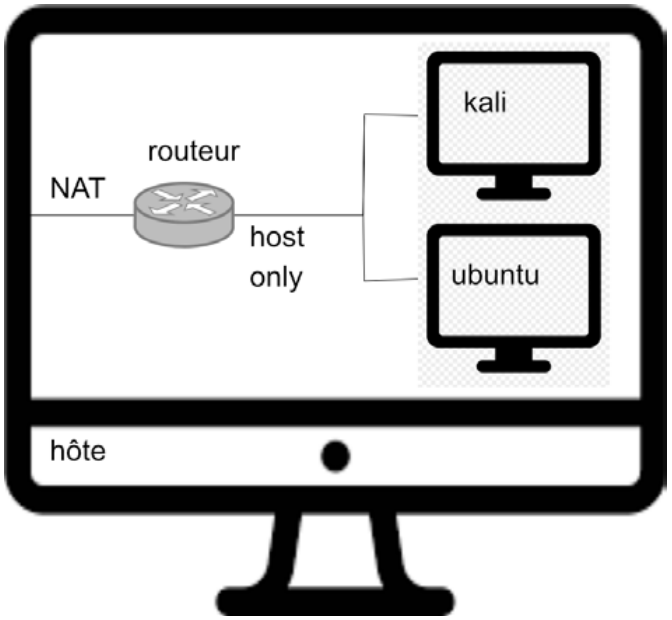
Connexion à l'ESXi : accès aux VM



Connexion à l'ESXi : accès aux VM



Configuration de cette année



Les machines !

A préparer pour le premier TP :

- LXLE [1.4Go download/ 8Go installed]
- Kali [2.4Go download/ 10Go installed] (kali/kali) avec la commande `localectl -no-convert set-x11-keymap fr` pour passer le clavier en français
- pfSense CE [2Go installed] qui ne sera utilisée qu'à partir du TP2

Je conseille l'utilisation de VMware workstation player dans sa version gratuite (pour l'éducation).

Les TP !

1. Apache2, `http`, `https`, MITM et `ufw`
2. `pfSense`, gestion routage, `dhcp`, `dns`
3. `postfix` et `imaps`, `ssh` et `gnuPG`
4. `openVPN` sur `pfSense`
5. `openVAS` et `Metasploit` ; audit de sécurité

Bibliographie


W. Diffie and M.E. Hellman.
New directions in cryptography.
IEEE Trans. on Inform. Theory, 22(6) :644–654, 1976.


D. Kahn.
La guerre des codes secrets.
InterEditions, 1980.


L.R. Knudsen.
Block ciphers – a survey.
In Springer Verlag, editor, State of the art in applied cryptography, number 1528 in LNCS, pages 18–48, 1998.


B. Martin.
Codage, cryptologie et applications.
Presses Polytechniques Universitaires Romandes, 2004.


D. Stinson.
Cryptographie, théorie et pratique.
International Thomson Publishing, 1995.