F2B505 : Surveillance de Réseaux pour la Sécurité

Sandrine Vaton

IMT Atlantique, INFO

Janvier 2019

Sécurité des Systèmes d'Information (SSI) (1/2)

Organisme : Entreprise, entité privée ou publique, service de l'Etat, etc...

Système d'Information

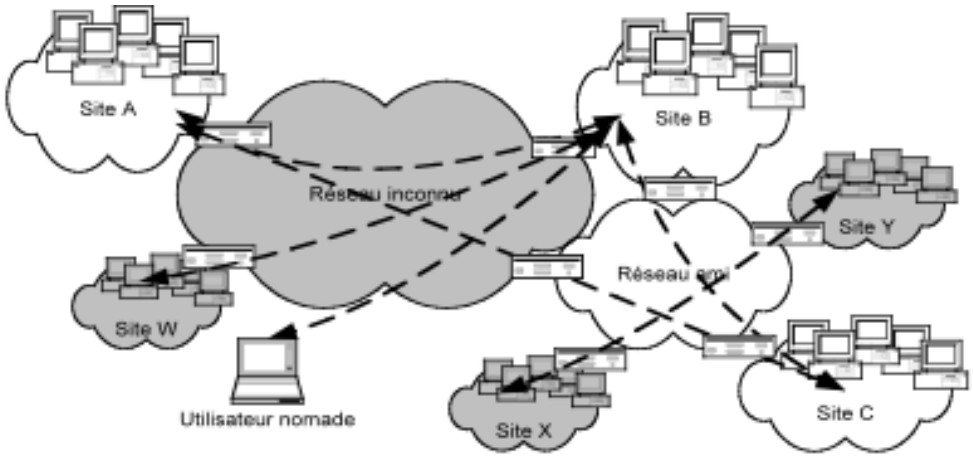
Ensemble des moyens (organisation, acteurs, procédures, systèmes informatiques, réseaux) :

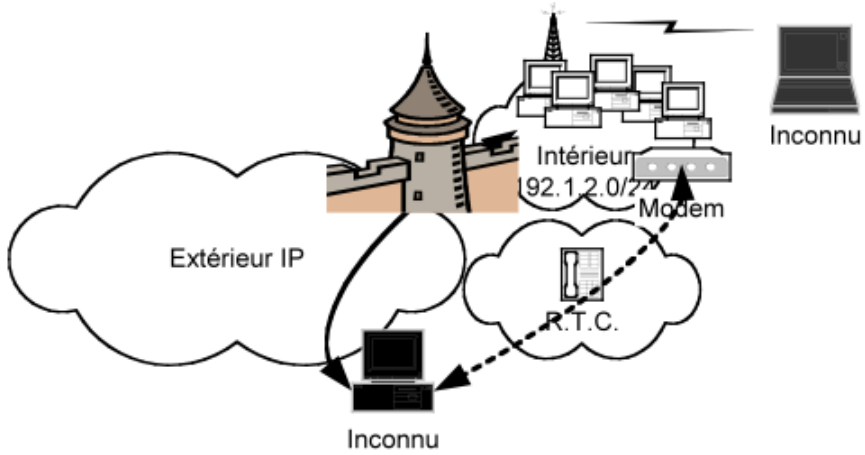
- **nécessaires au traitement et à l'exploitation des informations** (à partir des données),
- dans le cadre d'objectifs définis au niveau de la stratégie de l'organisme, des métiers, de la réglementation.

Sécurité des Systèmes d'Information (SSI) (2/2)

Pourquoi assurer la sécurité des SI ?

- Sites multiples, accès à l'Internet, accès extérieurs (à certains serveurs, applications métier)

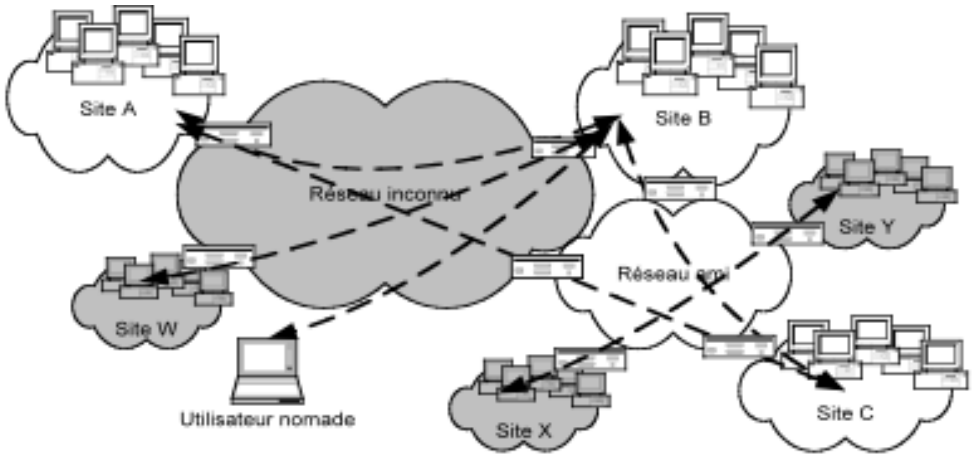




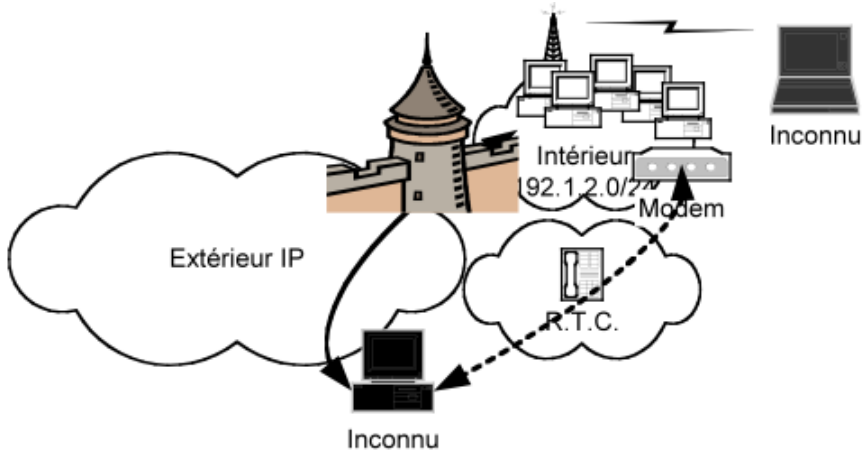
Sécurité des Systèmes d'Information (SSI) (2/2)

Pourquoi assurer la sécurité des SI ?

- **Standardisation** des machines et logiciels autour d'Internet
  $\rightarrow$ Quand une vulnérabilité est trouvée sur un système elle a de fortes chances d'être présente sur un système identique.



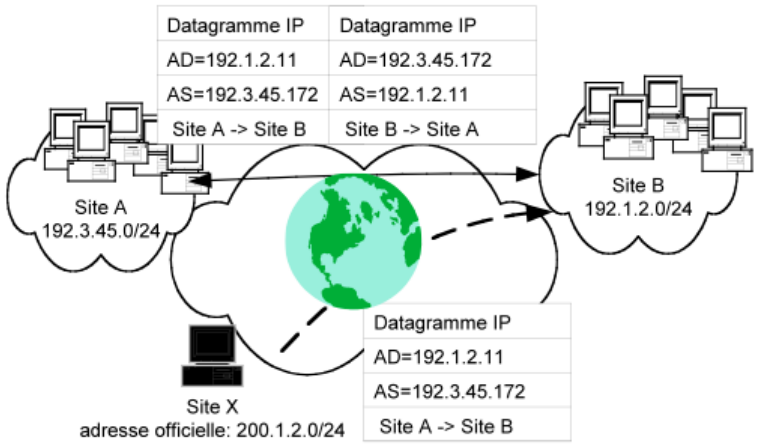
Temps de vie d'une machine Windows connectée à Internet sans antivirus ?



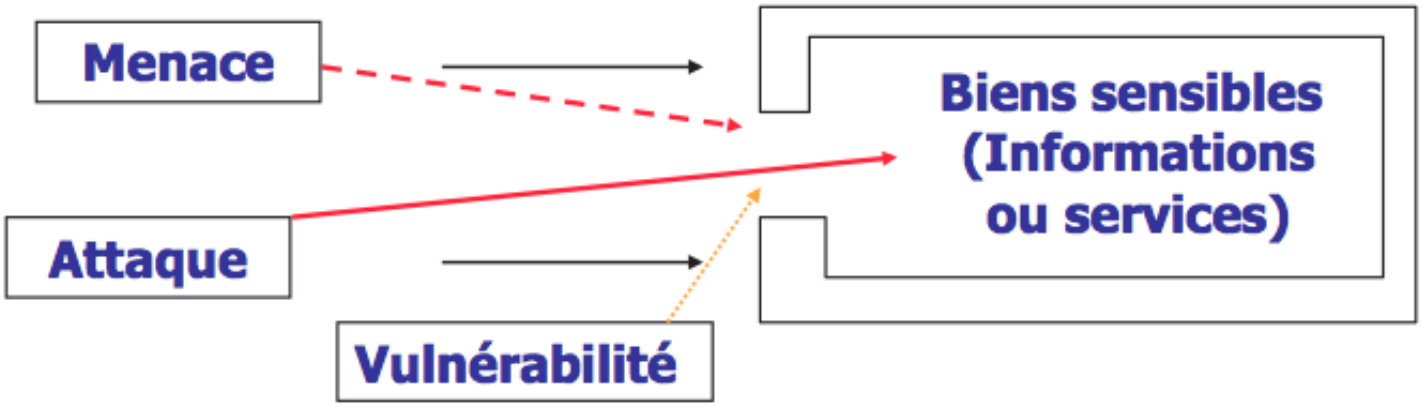
Sécurité des Systèmes d'Information (SSI) (2/2)

Pourquoi assurer la sécurité des SI ?

- Internet ne garantit pas l'authenticité d'une adresse IP
  - Facilite l'usurpation d'adresse IP = **IP spoofing**
  - Donne l'impunité à l'attaquant



Vulnérabilités, menaces, attaques



1. **Vulnérabilité :** Faute dans les spécifications, la conception, la réalisation, l'installation, la configuration d'un système, ou dans la façon de l'utiliser.
2. **Menace :** Risque qu'une vulnérabilité soit exploitée pour attenter à l'intégrité du système d'information.
3. **Attaque :** Action malveillante. Réalisation intentionnelle d'une menace exploitant une vulnérabilité.

Remarques : Une vulnérabilité peut être utilisée par un code d'exploitation et conduire à une intrusion dans le système. Connaître la vulnérabilité est la clé pour désigner la faille et la contre-mesure à appliquer.

Exemples de vulnérabilités, erreurs de conception

- **Buffer overflow** : une faille classique pour réaliser un exploit
  Principe : suite à une action du pirate, le programme écrit dans une zone mémoire où il n'était pas prévu qu'il aille écrire ; par exemple, s'il reçoit trop de données, le programme dépasse la limite prévue et écrit sur d'autres données, par exemple sur l'adresse de retour d'une fonction. Cela permet au pirate de lancer l'exécution d'instructions introduites par le pirate dans le processus.
- **Injections SQL** :
  Exploitation d'une faille de sécurité d'une application interagissant avec une base de données, en injectant une requête SQL non prévue par le système et pouvant compromettre sa sécurité. Le principe pour l'attaquant est de réussir à injecter dans la requête SQL des caractères non prévus pour modifier de manière malicieuse le comportement de la requête.

Exemples de vulnérabilités, erreurs de conception

- **Cross site scripting** (XSS) :
  Faille de sécurité des sites Web permettant d'injecter du contenu dans une page, permettant ainsi de provoquer des actions des navigateurs Web visitant la page (par ex. rediriger vers un autre site, ou voler une session en récupérant les cookies).
- **Elévation des privilèges** (*privilege escalation)* :
  Exploitation d'une faille de sécurité par un utilisateur lui permettant d'obtenir des privilèges supérieurs à ceux qu'il a normalement.
  L'utilisateur va pouvoir ainsi exécuter des tâches qui demandent des privilèges d'administrateur du système.

Comment savoir si un système est vulnérable ?

- **S'informer :** des organismes publient des avis de sécurité
  - informer la communauté des administrateurs sécurité de l'existence d'une vulnérabilité, et offrir une assistance aux traitements d'incidents
  - **CERT** (*Computer Emergency Response Team*) : centres d'alerte et de réaction aux attaques informatiques, à destination des entreprises et administration (information publique)

Exemples : CERT-FR (autrefois CERT-A, administration française) ;
CERT-LEXSI : CERT privé du groupe LEXSI, à destination des entreprises ;
CERT-RENATER : GIP Renater, réseau national enseignement recherche ;
CERT-Crédit Agricole ; CERT-Société Générale...
Exemple d'avis de sécurité : vulnérabilité dans OpenSSL (avril 2014) ("HeartBleed")

Référence	CERTFR-2014-ALE-003-002
Titre	Vulnérabilité dans OpenSSL
Date de la première version	08 avril 2014
Date de la dernière version	30 juillet 2014
Source(s)	Bulletin de sécurité OpenSSL du 07 avril 2014
Pièce(s) jointe(s)	Aucune

Taxonomie des logiciels et actions malveillantes (1/5)

- **Virus informatique :** automate auto-réplicatif, conçu pour se propager à d'autres ordinateurs en s'insérant dans des logiciels légitimes, et additionné de code malveillant Vecteurs de propagation : courriel électronique, clés USB... ; Parade : anti-virus, mise à jour permanente des bases de signatures de virus !
- **Ver informatique :** (*Worm*) logiciel malveillant qui se reproduit sur plusieurs ordinateurs via le réseau Internet. Contrairement au virus, un ver n'a pas besoin d'un programme hôte pour se reproduire. Objectif du ver : espionner l'ordinateur, détruire des données, offrir une porte dérobée à des pirates informatiques, participer à une attaque contre un serveur (en envoyant de multiples requêtes).
- **Logiciel espion** (*spyware*) : logiciel malveillant qui s'installe dans un ordinateur dans le but de collecter et transférer des informations sur l'environnement. Vecteurs de propagation : logiciels gratuits, certaines pages Web (exploitation de failles de sécurité du navigateur). But : profilage de la cible, par exemple pour lui envoyer de la publicité ciblée.

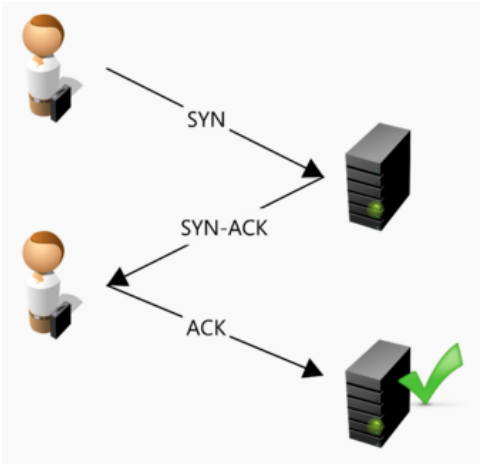
Taxonomie des logiciels et actions malveillants (2/5)

- **Cheval de Troie** (*Trojan Horse*) : logiciel en apparence légitime mais qui contient une malveillance. But : faire entrer le parasite contenu (virus, key logger, logiciel espion...) à l'intérieur de la cible.
- **Hammeçonnage** (*Phishing*) : technique reposant sur l'ingénierie sociale. Faire croire à la victime qu'elle s'adresse à un tiers de confiance (banque, ami, etc...) pour lui soutirer des informations personnels (numéro de carte de crédit, mot de passe).
  Exemple : pénétration du réseau Intranet de l'Elysée (probablement par la NSA) grâce à de faux "amis" sur Facebook
- **Autres** : enregistreur de frappe (*Key Logger*), publiciel (*adware*)

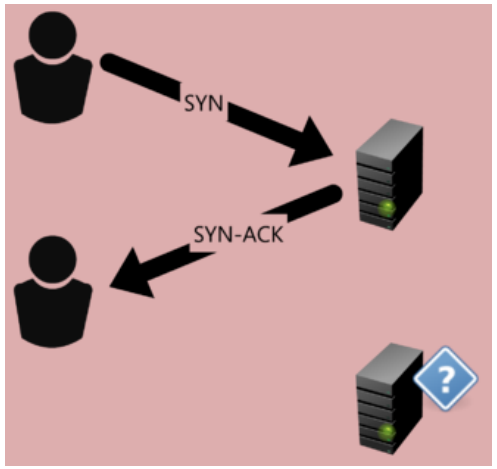
Taxonomie des logiciels et actions malveillants (3/5)

- **Déni de service** (*Denial of Service*, DoS) : type d'attaque visant à rendre indisponible un service pendant un certain temps.
  - Déni de service par saturation : submerger la cible de requêtes, afin qu'elle ne soit plus capable de répondre aux requêtes réelles.
  - Déni de service par exploitation de vulnérabilité : exploiter une faille du système distant afin de le rendre inutilisable.

Exemple : TCP SYN Flooding



3 Way Handshake



TCP SYN Flooding

Taxonomie des logiciels et actions malveillants (3/5)

- **Déni de service** (*Denial of Service*, DoS) : type d'attaque visant à rendre indisponible un service pendant un certain temps.
  - Déni de service par saturation : submerger la cible de requêtes, afin qu'elle ne soit plus capable de répondre aux requêtes réelles.
  - Déni de service par exploitation de vulnérabilité : exploiter une faille du système distant afin de le rendre inutilisable.

Autre exemple : Ping de la mort (*Ping of Death*).
Envoi de paquets Ping (ICMP echo request) malformés de façon à provoquer un crash de la machine cible. Paquets ping de taille supérieure à 56 octets non supportés par certains OS jusque dans les années 1997-98.

Déni de Service (1/2)

- **Déni de service distribué** (*Distributed Denial of Service*, DDoS) :
  - en règle générale les attaques de DoS sont distribuées, c'est-à-dire qu'elles sont mises en oeuvre par un grand nombre de machines simultanément, de façon à générer une charge suffisante pour faire tomber la cible
- **Réseau de machines zombies**(*Botnet*)
  - les attaques de DDoS sont en général réalisées par un réseau de machines compromises contrôlées à distance par un hacker (*bot herder*)
  - Mode opératoire :
    - ★ scanner le réseau pour détecter des vulnérabilités
    - ★ exploiter ces vulnérabilités pour installer des *bots* (robots logiciels) sur les machines qui deviennent ainsi compromises (*zombies*)
    - ★ contrôler à distance les zombies pour lancer des attaques de manière synchronisée contre une cible
  - Communication entre le *bot herder* et les *zombies* : souvent par canal IRC (ou alors en P2P)

Déni de Service (2/2)

- **Attaques par rebond**
  - exploite la possibilité pour une machine d'usurper l'adresse IP de la victime (adresse IP source "spoofée", *spoofed Src IP @*)
  - une victime, des machines zombies (*bots*) utilisées pour lancer l'attaque, d'autres machines (a priori non corrompues) utilisées pour amplifier l'attaque
  - Mode opératoire :
    - ★ chaque zombie envoie à des machines utilisées comme amplificateurs des requêtes avec comme adresse IP source celle de la victime (par ex. TCP SYN)
    - ★ les machines amplificatrices répondent *à la victime* (TCP SYN ACK)
    - ★ la victime subit alors un DDoS
  - "intéressant" lorsque le volume des réponses est beaucoup plus important que la taille des requêtes elles-mêmes ;
    exemple : amplification par utilisation du *DNS* (contre Spamhaus, mai 2013, 300 Gb/sec) ; amplification utilisant *NTP* (février 2014, 400 Gb/sec)

Résilience de l'Internet

- attaquer le serveur web d'une entreprise (ex : défiguration-"defacing") a un impact psychologique mais ne constitue une réelle "menace" que pour l'image de cette entreprise, les usagers de ce service
- attaquer les **infrastructures de l'Internet** permet de toucher potentiellement plus de monde
- **Résilience :** capacité à fonctionner normalement, robustesse, capacité à limiter les impacts d'un incident
  voir Dossier AFNIC, *Peut-on casser l'Internet ?* ; pour résumer : "l'Internet est localement vulnérable et globalement robuste" (Pierre Col)
  Résilience de l'Internet français 2016, rapport publié par l'ANSSI (Agence Nationale de Sécurité des Systèmes d'Information) et l'AFNIC (Association Française pour le Nommage Internet en Coopération)
- Deux protocoles essentiels au fonctionnement de l'Internet :
  - **BGP** (*Border Gateway Protocol*), qui permet d'acheminer les données à l'aide d'annonces de routage
  - **DNS** (*Domain Name System*), qui fournit la correspondance entre un nom de domaine et une adresse IP

Attaques contre le DNS

Empoisonnement (ou Pollution) de cache DNS (*DNS cache poisoning*)

- DNS (*Domain Name System*) : service permettant de traduire un nom de domaine en diverses informations, en particulier une adresse IP (telecom-bretagne.eu : 192.44.76.253)
- Empoisonnement de cache DNS :
  - leurrer les serveurs DNS afin de leur faire croire qu'ils reçoivent une réponse valide à une requête qu'ils effectuent, alors qu'elle est frauduleuse.
  - Une fois le serveur DNS empoisonné, l'information est mise en cache, rendant ainsi vulnérable tous les utilisateurs de ce serveur.
  - Permet d'envoyer un utilisateur vers un faux site (hameçonnage, vecteur de virus ou autres applications malveillantes).
  - Un ordinateur s'adresse normalement au serveur DNS de son fournisseur d'accès. DNS : système hiérarchique et distribué $\Rightarrow$ les informations erronnées peuvent se propager à d'autres serveurs DNS (esclaves) et toucher ainsi un grand nombre d'utilisateurs.

Parade : déployer DNSSec (IETF, RFC 4033) qui permet de signer cryptographiquement les enregistrements DNS. Bien répartir les serveurs DNS faisant autorité

Attaques contre le DNS

Empoisonnement (ou Pollution) de cache DNS (*DNS cache poisoning*)

- DNS (*Domain Name System*) : service permettant de traduire un nom de domaine en diverses informations, en particulier une adresse IP (telecom-bretagne.eu : 192.44.76.253)
- Empoisonnement de cache DNS :
  - leurrer les serveurs DNS afin de leur faire croire qu'ils reçoivent une réponse valide à une requête qu'ils effectuent, alors qu'elle est frauduleuse.
  - Une fois le serveur DNS empoisonné, l'information est mise en cache, rendant ainsi vulnérable tous les utilisateurs de ce serveur.
  - Permet d'envoyer un utilisateur vers un faux site (hameçonnage, vecteur de virus ou autres applications malveillantes).
  - Un ordinateur s'adresse normalement au serveur DNS de son fournisseur d'accès. DNS : système hiérarchique et distribué $\Rightarrow$ les informations erronnées peuvent se propager à d'autres serveurs DNS (esclaves) et toucher ainsi un grand nombre d'utilisateurs.

Autres attaques : surveillance, censure (DNS "menteurs"), pannes...

Attaques contre BGP (1/2)

- **BGP** (*Border Gateway Protocol*)
  - un opérateur gère des blocs d'adresses IP contigues (appelés préfixes)
  - les opérateurs s'interconnectent à l'aide de BGP (IETF, RFC4271) qui permet d'échanger des informations d'accessibilité (préfixes) entre Autonomous Systems (AS)
  - les interconnexions entre deux AS se divisent en deux catégories (peering, transit)
    - ★ peering : chaque AS (pair, *peer*) annonce à l'autre les préfixes qu'il gère ; tout le trafic entre deux pairs est échangé directement
    - ★ transit : le client (société, fournisseur d'accès, diffuseur de contenu...) annonce les préfixes qu'il gère à son opérateur de transit ; l'opérateur de transit propage l'annonce du client, et en retour lui annonce le reste des préfixes constituant l'Internet.

Attaques contre BGP (2/2)

- **Usurpation d'annonces BGP**
  - annoncer des routes pour des adresses IP autres que les siennes
  - capter le trafic de sa victime pour la couper du réseau (déni de service), ou se faire passer pour sa victime (et, par exemple, lire son courrier)
  - très rapidement, la victime ne reçoit quasiment plus de trafic
  - *shunt BGP :* réinjecter le trafic à la victime après l'avoir écouté ou modifié, de façon à retarder la détection de l'attaque ; principe : détourner un préfixe IP depuis tout l'Internet, tout en maintenant un *clean path* pour le retour (*Man In the Middle*)
- **Parades :** BGPSec, en cours de définition à l'IETF ; objectif : permettre d'authentifier les routes
  - RPKI, infrastructure de gestion de clés (IGC) dédiée à la certification des ressources IP (préfixes, numéros d'AS) ; organismes certificateurs enregistrés auprès de l'IANA
  - *resource certificates* : certificats X509 attestant qu'un membre détient des préfixes IP et des numéros d'AS
  - ROA (*Route Origin Authorization*) : ce sont des objets route signés

Motivations – Evolutions

Avec le développement d'Internet, les attaques sont de plus en plus **massives et sévères**.
Exemple : le ver Conficker, apparu fin novembre 2008, aurait infecté de l'ordre de 9 millions d'ordinateur en janvier 2009.

Les motivations derrière les attaques informatiques ont évolué, du simple exploit (début des années 90) à l'extorsion de fonds, au vol de données (comptes, passeports...), au **crime organisé** et à la **cyber-guerre**.

Exemples d'attaques

- ver **Conficker** (2008-2009) qui a infecté 7 millions d'ordinateur Windows en quelques mois. N'exploite que des vulnérabilités connues, défaut de mise à jour des OS, logiciels. Beaucoup d'institutions infectées : Intramar (intranet de la marine nationale française), département de la défense US, ministère de la défense UK...
- **Stuxnet**, ver informatique affectant les systèmes Windows, supposément développé par les USA et Israel. A infecté en 2010, 45000 systèmes informatiques dont 30000 en Iran. Premier ver visant à espionner et reprogrammer des systèmes industriels (systèmes SCADA produits par Siemens pour contrôler, par exemple, des centrales nucléaires).

Exemples d'attaques

- attaque de phishing, pénétration du réseau Intranet de l'**Elysée** en 2012 (probablement, NSA) ;
- nombreuses **attaques de DDoS**, par exemple contre Paypal par les Anonymous suite à la fermeture du site MegaUpload
- affaires de **faux certificats de clés publiques** ; par exemple faux certificats Microsoft émis par la PKI Diginotar ;
- atteintes à la résilience de l'Internet ; ex : **blocage de l'accès Internet** pendant le printemps arabe (exemple : Libye 2011, source : Google transparency report)

Exemples d'attaques

- Exemples d'incidents de routage dûs à des **annonces BGP illégitimes** :
  - 2008, Pakistan Telecom annonce à son opérateur de transit des préfixes plus spécifiques que ceux utilisés par YouTube ; propagation de ces préfixes à l'ensemble de l'Internet, et **redirection d'une partie du trafic YouTube vers Pakistan Telecom**
  - 2010, **China Telecom** annonce plusieurs dizaines de milliers de préfixes ne lui appartenant pas, redirection du trafic à destination de ces préfixes
  - Février 2013, **détournement du trafic** de différentes victimes (institutions financières, gouvernements, ISP) **vers GlobalOneBel** (opérateur biélorusse) sans interruption du trafic ; attaque observée par Renesys ; en mai 2013, **détournement de trafic vers l'Islande**...

Exemples d'attaques

Exemple d'attaque par amplification : utilisation du DNS

- Amplification par utilisation du DNS – Mode opératoire :
  - envoi de requêtes DNS avec une adresse IP Source spoofée (adresse de la victime)
  - demande de tous les enregistrements d'une zone volumineuse (ex : ripe.net) ; requête : quelques dizaines d'octets, réponse : plusieurs kilo-octets (facteur 100) $\Rightarrow$ saturation des liens d'accès vers les victimes
  - Rem : DNS utilisant UDP, pour bloquer le trafic (au niveau d'un firewall) il faut analyser le contenu de la payload (charge utile du paquet) et pas seulement les entêtes $\Rightarrow$ lourd !
  - exploite l'existence de serveurs DNS configurés en OpenResolver $\rightarrow$ résolution récursive sans filtrage des clients (clients en dehors du domaine administratif du serveur DNS)
- violente attaque en mai 2013 contre Spamhaus (société établissant des blacklists d'adresses expédiant du spam), pics à 300 Gb/sec de trafic d'attaque

Exemples d'attaques

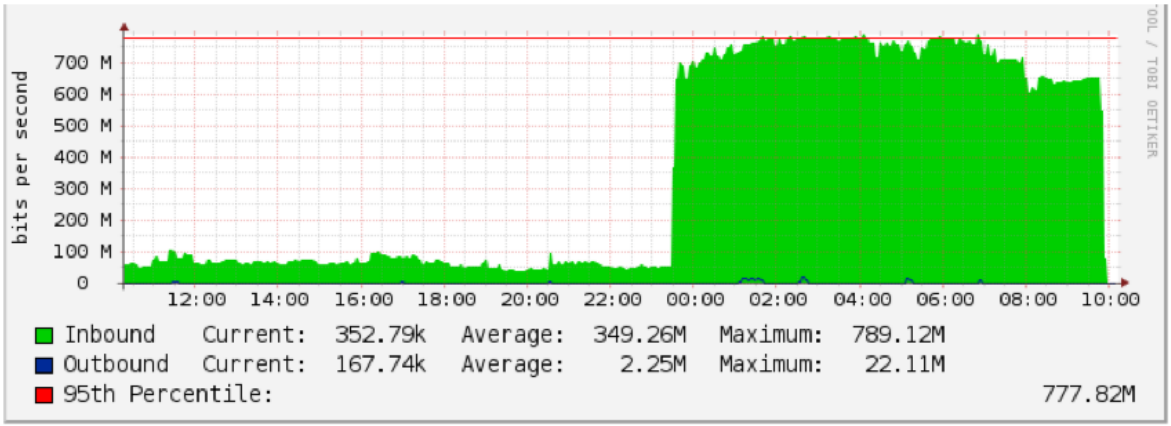
Exemple d'attaque par amplification : utilisation du DNS

- Mitigation de l'attaque contre Spamhaus :
  - utilisation de l'*anycast CDN*
    - ★ CDN : Content Delivery Network ; réseau de distribution de contenu
    - ★ anycast : rediriger les requêtes vers le serveur le plus proche d'un client
  - CloudFare : plusieurs datacenters annonçant chacun la même adresse IP
  - les serveurs DNS étant répartis sur l'ensemble du globe, les réponses aux requêtes sont envoyées sur les différents points de présence du CDN CloudFare
- Voir sur le CERT-IST : `http://www.cert-ist.com/public/fr/SO_detail?code=201304_spamhaus`

Exemples d'attaques

Autre exemple d'attaque par amplification : utilisation du NTP

- voir par exemple le billet sur le blog de Stephan Bortzmeyer : `http://www.bortzmeyer.org/ntp-reflexion.html`
- Exemple de trafic pendant une attaque NTP Amp :



Exemples d'attaques

Attaques "récentes"

- **attaque contre DynDNS** :
  - 21/10/2016, attaque DoS massive contre le service de résolution DynDNS (trafic : plus de 1 To/sec)
  - nombreux sites utilisant le service Dyn Managed DNS2 inaccessibles de 7h00 à 17h00 UTC : Twitter, Ebay, Netflix, GitHub, PayPal, sont inaccessibles pendant une dizaine d'heures (de 7h00 à environ 17h00 UTC)
  - utilisation par les attaquants d'objets connectés piratés (caméras de surveillance, baby phones...)
- **Amazon Route 53 BGP Hijacking :**
  - fin avril 2018 : détournement de trafic par BGP hijacking et fausses annonces DNS pour voler des crypto-monnaies ; vols d'ethers pour un montant de 20 millions d'euros
  - détournement d'une partie du trafic Internet des usagers de MyEtherWallet.com vers un serveur russe
  - manipulation des requêtes DNS envoyées vers Route 53, un service d'Amazon Web Services
  - interception DNS faite en manipulant le protocole BGP ; fausses informations de routage relatives au service Amazon Route 53

Et les pannes...

- les problèmes de disponibilité ne sont pas dûes, le plus souvent, à des attaques, mais à des **pannes d'équipements** ou des **erreurs de configuration**
- Exemples :
  - coup de pelleteuse : coupure de fibre optique et impact sur un FAI, vols de cables en cuivre qui prive un village rural de son accès Internet...
  - pannes sur les équipements d'un PON (Passive Optical Network) qui impacte certains abonnés d'un réseau FTTH
  - à plus grande échelle, Internet, panne au niveau d'un IXP (Internet Exchange Point) comme Amsterdam IX en 2016 : indisponibilité de certaines routes, dégradation de QoS

Métrologie des réseaux

- nécessaire **surveillance continue du réseau** :
  - **surveiller le réseau** de manière continue, pour **détecter** des anomalies (si possible avant appel à la hotline !) et les **localiser** (trouver une cause racine) ; souvent, analyses post-mortem
  - différentes **métriques** à surveiller, suivant qu'on est un réseau local, un FAI, Internet...
  - surveiller l'**état des équipements** ; ex : informations remontées par les équipements d'un PON (Eb/NO, température, intensités/tensions, alarmes intermédiaires...)
  - surveiller les **performances du réseau** sur Internet (RTT...), les routes (traceroute...) ; NB : on parle ici de métrologie active (ping...)
  - Internet : surveiller le routage externe, les annonces BGP
  - Internet : surveiller la disponibilité du **service DNS**
  - surveiller le **trafic**, différents niveaux de granularité possibles
    - ★ très gros grain : **volumes de trafic** au niveau des différentes interfaces (requêtes SNMP aux MIB des routeurs) ;
    - ★ à un grain plus fin : numéros de ports, volumes de trafic par application (statistiques de niveau flot avec **NetFlow**) ;
    - ★ à un grain très fin, surveillance de niveau du paquet (*Wireshark*) et monter plus ou moins haut dans les couches protocolaires

Métrologie des réseaux

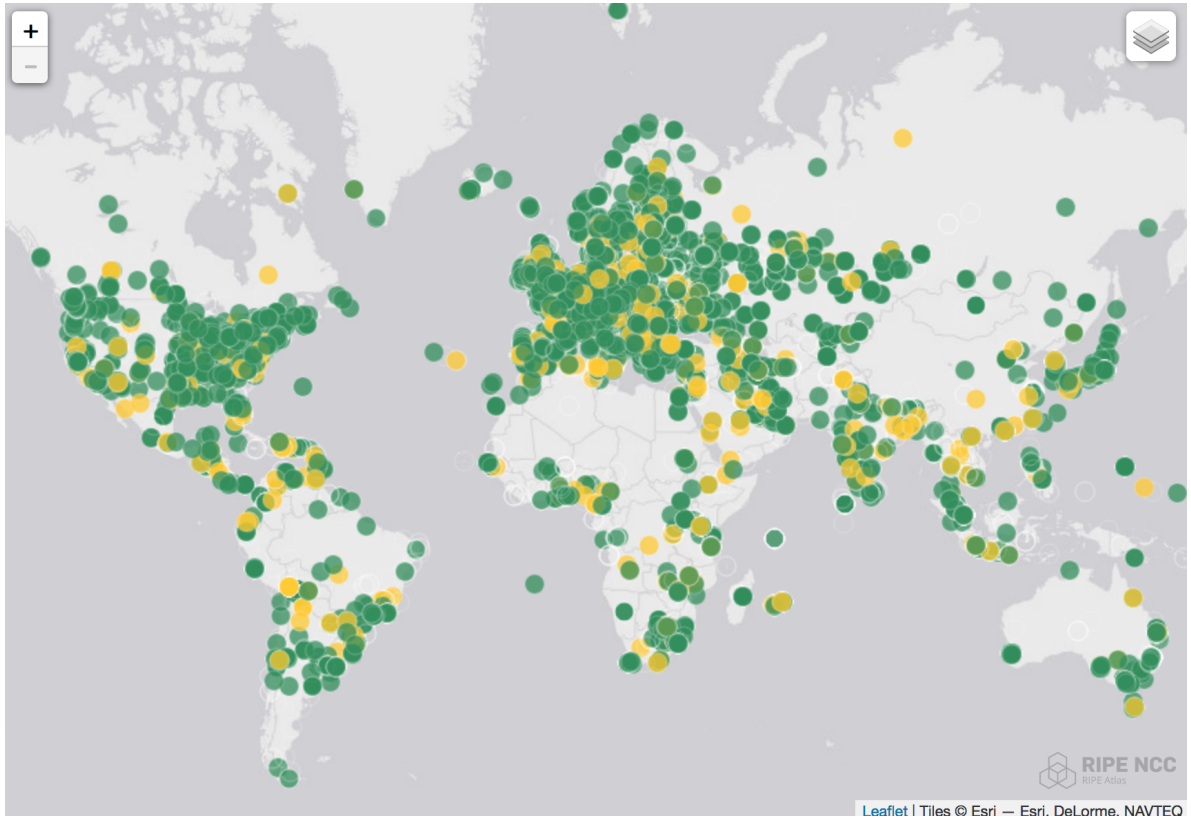
- **métrologie des réseaux** – différentes approches possibles :
  - taxonomie pour la détection d'anomalies
    - ⋆ approches **basées signatures** (base de signatures connues des causes de pannes les plus classiques)
    - ⋆ approches **basées anomalies** (détecter un changement par rapport au fonctionnement habituel du réseau)
  - techniques d'**analyse de données** :
    - ⋆ analyse de **séries temporelles**, détection de **changement brusque**, techniques de **comptage** sur des flux en temps réel (*data stream mining*), **classification/clustering**, **modèles statistiques** plus ou moins complexes...
    - ⋆ **capture et analyse** de trafic à haut débit avec des approches logicielles (DPDK, jusqu'à $N \times 10$ Gb/sec) ou accélération matérielle (FPGA, par exemple NetFPGA)
    - ⋆ analyse de **gros volumes de données** (big data) : Hadoop, Spark, Apache Storm...

RIPE Atlas

- **RIPE Atlas** : plateforme de mesure de l'Internet, ouverte, distribuée et à l'échelle mondiale ; `https://atlas.ripe.net/`
- projet collaboratif de métrologie de l'Internet piloté par **RIPE NCC** (Réseaux IP Européens, registre Internet régional, RIR, pour Europe/Asie de l'Ouest/URSS)
- **sondes installées** par des volontaires (souvent, des particuliers) dans le monde entier ; **ancres** hébergées dans des endroits ayant une bonne connectivité Internet (opérateurs de DNS, hébergeurs, universités, etc...)
- les sondes sont contrôlées à distance et effectuent des mesures ; différents types de mesures : **ICMP echo (ping), traceroutes, requêtes DNS, HTTPS...**
- **mesures routinières** entre les ancres ; possibilité pour un usager de **configurer sa propre campagne** de mesure (système de crédits)
- résultats des mesures en **accès public** ; donne une forme de visibilité sur l'"état de l'Internet" à une échelle globale, surveillance distribuée du réseau

RIPE Atlas

Une sonde Atlas



Couverture du réseau RIPE Atlas (jaune : ancres, vert : sondes) ; environ 10 000 sondes connectées en 2019, beaucoup en Europe et Amérique du Nord, moins dans le reste du monde

Système de détection des intrusions

Qu'est-ce qu'un IDS (Intrusion Detection System) ?

- La prévention fonctionne lorsque :
  - les utilisateurs internes du système sont des personnes de confiance
  - il y a peu d'interactions avec l'extérieur (accès réseau, clés USB, disques externes...)
- Besoin d'un système qui vienne compléter la prévention

Intrusion Detection System

Un système de détection d'intrusions (IDS) est un dispositif logiciel ou matériel utilisé pour détecter les tentatives d'accès non autorisées à un ordinateur ou à un réseau.

Système de détection des intrusions

Pourquoi un IDS ?

- La sécurité c'est avant tout la PREVENTION
  - Protection physique au niveau hardware
  - Mots de passe et autres moyens d'*authentification*
  - Listes d'accès pour gérer les *autorisations*
  - Cryptographie pour la *confidentialité*
  - Sauvegardes (backup) pour la *restauration de données*
  - Filtrage (firewalls) pour limiter le trafic à celui autorisé
  - Sensibilisation des utilisateurs (*ingénierie sociale*, maladresses...)
  - etc... ...

MAIS ...

... la sécurité absolue n'est JAMAIS garantie !

Système de détection des intrusions

Différents types d'"intrusions"

- **EavesDropping (écoutes indiscrètes, "sniffing" du trafic)** : interception passive du trafic sur une interface réseau (outils : TCPdump, WireShark)
- **Snooping and Downloading** : récupération d'informations personnelles comme des emails, des données business, des mots de passe... (outils : keyloggers, écoute sur réseau WiFi...)
- **Tampering** : attenter à l'intégrité des données ; modification malicieuse de données, de logs...
- **Spoofing** : vol d'identité, se faire passer pour quelqu'un d'autre ; exemple : IP Spoofing (modification de l'adresse source des paquets IP)
- **Flooding, Denial of Service** : faire écrouler un système en le submergeant de requêtes ; exemple : TCP SYN flooding
- **Phishing** (hameçonnage) : ingénierie sociale, exploiter la naïveté d'utilisateurs autorisés (pour les amener sur un site malicieux) ; **Hijacking** (détournement) (ex : détournement du trafic à des fins malicieuses)
- **Craquage de mots de passe et de clés**
- ...

Système de détection des intrusions

Différents types d'"intrus" *(J.P.Anderson, Computer Security Threat Monitoring and Surveillance)* :

- **External penetrator** : un individu non autorisé à utiliser un ordinateur qui réussit d'une façon ou d'une autre à contourner les mécanismes de contrôle d'accès
- **Masquerador** : intrus, interne ou externe, qui a pu avoir accès au système en utilisant l'authentification d'un utilisateur autorisé
- **Misfeasor** : utilisateur légitime qui outrepasse ses droits pour accéder à des données, programmes, ressources pour lesquelles l'accès ne lui est normalement pas autorisé
- **Clandestine User** : type d'intrus difficile à détecter ; a réussi à accéder au système avec un très haut niveau de privilèges et à supprimer les enregistrements d'audit des actions qu'il vient d'accomplir

Système de détection des intrusions

Qui est vulnérable aux intrusions ?

- **Institutions financières, banques** ... Exemple : attaques contre PayPal par les Anonymous (DDoS) suite à la fermeture de MegaUpload
- **Institutions gouvernementales** ... Exemple : intrusion réussie sur l'Intranet de l'Elysée (hameçonnage), attaque (DoS) contre le site du Sénat (site de la députée UMP Valérie Boyer, proposition de loi sur les génocides dont l'Arménie)
- **Sites industriels**... Exemple : perturbation de systèmes de commande industrielle (Iran)
- **Fournisseurs d'accès**
- **Fournisseurs de services** (Over The Top OTT ; YouTube, FaceBook...)
- **Multinationales**
- TOUT LE MONDE !

Taxonomie des IDS

Les IDS peuvent être classés sur la base de différents critères :

1. **Système surveillé**
   - IDS hôte (Host IDS, HIDS)
   - IDS réseau (NIDS)
2. **Architecture**
   - Centralisée
   - Distribuée
3. **Techniques d'analyse**
   - Stateful
   - Stateless
4. **Techniques de détection**
   - IDS basé signature
   - IDS basé anomalies

Taxonomie des IDS

IDS hôte vs. IDS réseau

IDS hôte

- A pour objet de détecter des tentatives d'intrusion au niveau d'un hôte particulier
- Dépend de l'architecture de machine et du système d'exploitation utilisé
- Traite des données haut niveau (par ex. des appels système)
- Utile pour détecter des intrusions venant de l'"intérieur"

IDS réseau

- A pour objet de détecter des attaques à destination de machines sur un réseau local
- Dépend des architectures de machines et des systèmes d'exploitation
- Traite des données à un niveau de granularité très fin (paquets IP)
- Utile pour détecter des attaques venant de l'"extérieur"

Taxonomie des IDS

IDS centralisé vs. IDS distribué

IDS centralisé

- Toutes les opérations sont réalisées au niveau d'un seul point
- Plus simple à réaliser
- Un seul point de dysfonctionnement

IDS distribué

- Constitué de plusieurs composants
  - **Capteurs** qui génèrent des événements de sécurité
  - **Console** pour surveiller les événements et les alertes et contrôler les capteurs
  - **Moteur** central qui enregistre/corrèle les événements et génère les alarmes
- Besoin de travailler avec différents formats de données
- Standards :
  - IPFIX pour remonter des mesures de niveau flot des points de capture au collecteur centralisé
  - IDMEF pour échanger des alarmes

Taxonomie des IDS

Stateless IDS vs. Stateful IDS

Stateless IDS

- Traite chaque événement indépendamment des autres
- Simple à concevoir
- Passe à l'échelle en termes de vitesse de traitement

Stateful IDS

- Maintient une information sur les événements précédents
- L'interprétation d'un événement dépend de sa position dans un flot d'événements
- Plus complexe à concevoir
- Traitements lourds, problèmes potentiels de passage à l'échelle

Taxonomie des IDS

IDS basés signature vs. IDS basés anomalie

IDS basés signature

- Identifient les intrusions en reconnaissant des patterns spécifiques supposément malicieux ("signatures") dans le trafic ou dans les données applicatives
- Base de données de signatures devant être mise à jour régulièrement
- Ne peuvent détecter que des attaques “connues” ; problèmes avec les attaques "zero day"

IDS basés anomalie

- Identifient les intrusions en détectant des comportements inhabituels
- Nécessitent une phase préalable de calibration pour apprendre les comportements normaux
- Capables de détecter des attaques "nouvelles"
- Génèrent plus de fausses alarmes que les IDS basés signature

Générations d'IDS

On peut distinguer trois grandes époques dans l'évolution des IDS

1. IDSs de Première Génération (fin des années 1970s)
   - Apparition du concept d'IDS à la fin des années 1970s, début des années 1980s (Anderson, *Computer Security Monitoring and Surveillance*, Tech Rep 1980)
   - Audit de données sur **une seule machine**
   - Traitement hors ligne des données (**post processing**)
2. IDSs de Seconde Génération (1987)
   - Intrusion Detection Expert System (Denning, *An intrusion Detection Model*, IEEE Trans. on Soft. Eng., 1987)
   - **Analyse statistique** de données
3. IDSs de Troisième Génération (en cours)
   - **IDS réseau**
   - Détection et réaction en **temps réel**
   - Intrusion Detection System $\Rightarrow$ **Intrusion Prevention System**

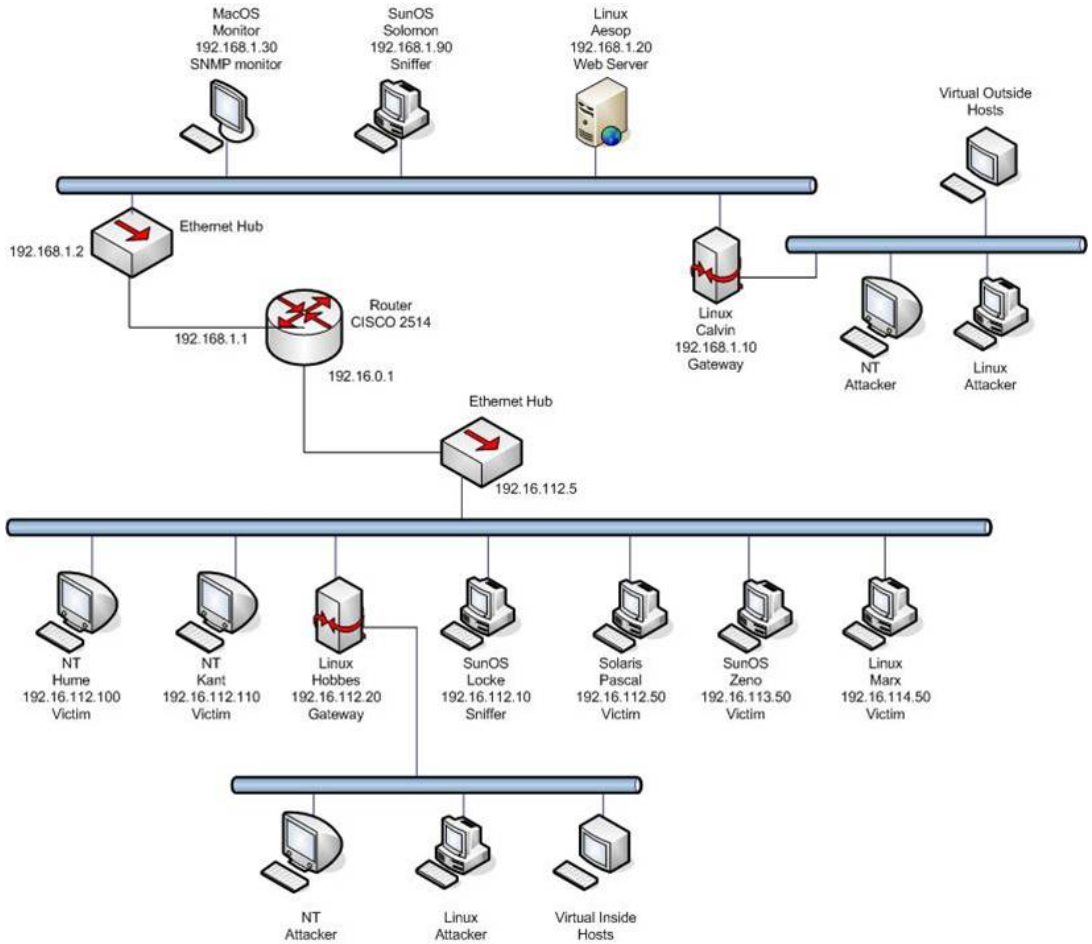
Programmes d'évaluation/ Jeux de données de test

Programme d'évaluation DARPA

- 1998/1999 DARPA/MIT IDS evaluation program : programme complet d'évaluation des IDS (le plus complet à ce jour)
- Fournit un grand nombre de données pour le développement, l'amélioration, l'évaluation des IDS
- Différents types de données disponibles :
  - logs au niveau système
  - trafic réseau
- Simulations de trafic réseau (entre une base de l'US Air Force et l'Internet) pour la génération du trafic normal ; attaques expérimentales

Programmes d'évaluation/ Jeux de données de test

Le réseau DARPA



Programmes d'évaluation/ Jeux de données de test

Dataset DARPA/MIT

- 5 semaines de données
  - semaines 1 et 3 : pas d'attaques, utilisables pour l'entraînement du système
  - semaine 2 : attaques étiquetées ; peut être utilisé pour construire des bases de signature
  - semaines 4 et 5 : différentes attaques ; peut être utilisé pour évaluer les performances en détection des IDS
- Différents types d'attaques : Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), ... ; 177 instances de 59 types différents d'attaques
- La vérité sur les attaques est fournie ("Attack Truth list") ce qui permet d'évaluer le taux de fausse alarme (faux positifs) et de non détection (faux négatifs) des IDS testés

MAIS... le jeu de données DARPA est considéré comme *obsolète* car trop ancien pour refléter la nature actuelle du trafic et des attaques.

Programmes d'évaluation/ Jeux de données de test

Autres datasets KDD99 :

- autre trace publique *étiquetée* disponible pour l'évaluation d'IDS
- créée spécialement pour l'évaluation d'IDS lors du « Third International Knowledge Discovery and Data Mining Tools Competition »
- comme DARPA, KDD99 est jugée obsolète

Autres données publiques :

- il existe différentes traces de trafic accessibles publiquement ; e.g. CAIDA, Abilene (Internet2), GEANT, ...
- problème majeur : pas de "ground-truth" !

D'une manière générale la recherche dans le domaine a besoin d'accéder à des mesures qui soient à la fois dans le domaine public et représentatives
$\Rightarrow$ Problème récurrent, difficultés liées à la confidentialité des données opérateurs

L'IDS Snort

Qu'est-ce-que Snort ?

SNORT : NIDS basé signatures

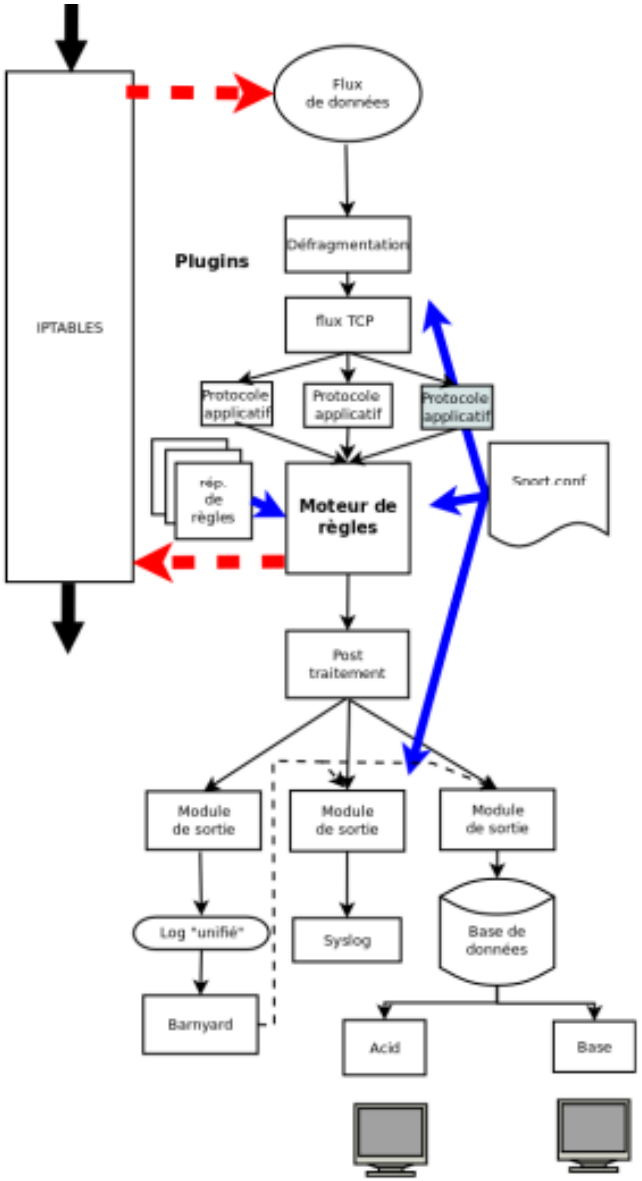
- Snort est un système de détection d'intrusion réseau (**NIDS**).
- Snort est un IDS basé sur la **signature** de l'attaque (basé "scénarios")
- Snort est **open-source** à la fois pour le code source et pour les règles.

Modes d'utilisation de Snort

- Mode **Sniffer** : "renifleur" de paquet où il se comporte comme tcpdump.
- Mode **Logger** : enregistreur de paquet : utile pour garder des traces (dans des buts d'enquête légale, pour confirmer des hypothèses...)
- Mode **IDS** classique.
- Mode en ligne **IPS**. Il s'appuie sur iptables (le firewall) pour intercepter les paquets et pour interrompre les flux.

L'IDS Snort

Fonctionnement de Snort



- Les **plugins** préparent le flux pour l'analyse et contrent les techniques classiques d'évasion.
- Les **règles** sont le coeur de l'analyse.
- Des modules de post traitement réalisent des gestions de la sortie et des filtrages supplémentaires.
- Les **modules de sortie** mettent en forme et envoient les résultats pour traitement.

L'IDS Snort

Format des règles Snort :

Les **règles Snort** sont décrites dans un langage simple et suivant le schéma suivant :

- l'en-tête de la règle qui contient :
  - l'action de la règle (la réaction de Snort)
  - le protocole qui est utilisé pour la transmission des données (Snort en considère trois : TCP, UDP et ICMP)
  - les adresses IP source et destination et leur masque
  - les ports source et destination sur lesquels il faudra vérifier les paquets.
- les options de la règle (entre parenthèses) qui contiennent :
  - le message d'alerte
  - les conditions qui déterminent l'envoi de l'alerte en fonction du paquet inspecté.

L'IDS Snort

Exemple simple de règle Snort

Cette règle détecte les tentatives de login sous l'utilisateur root, pour le protocole ftp (port 21) :

Exemple :

alert tcp any any -> 192.168.1.0/24 21 (content : "USER root" ; nocase ; msg : "Tentative d'accès au FTP pour l'utilisateur root" ;)

Les messages en direction de cette plage d'adresse IP effectuant une tentative de login root ("USER root" contenu dans le paquet) auront pour conséquence la génération de l'alerte "Tentative d'accès au FTP pour l'utilisateur root".

L'IDS Snort

Les sorties de Snort :

Format de la sortie

Snort génère deux types de sorties :

- une sortie d'alertes contenant au moins l'identifiant complet de la règle, les sources et destinations impliquées et la date
- en parallèle un enregistrement des paquets impliqués dans l'alerte.

Un identifiant est défini par un groupe (module), un identifiant de règle, et l'analyse ainsi qu'un numéro de révision.

Gestion de la sortie

La sortie peut être :

- envoyée dans des fichiers, dans une base de données, passée directement à syslog, etc
- stockée dans un format intermédiaire pour laisser la charge du traitement à un autre processus

L'IDS Snort

Les actions possibles

- Les actions normales :
  - **alert :** génère une alerte puis log
  - **log :** enregistre le paquet
  - **pass :** ignorer le paquet
  - **activate :** alerte et alors active une autre règle dynamic
  - **dynamic :** reste passive jusqu'à être activée par une règle activate, alors agit comme une règle log
- Les actions IPS :
  - **drop :** le paquet est perdu par iptables et un log est fait.
  - **reject :** le paquet est perdu, un log est créé puis on envoie un RESET des listes de port (TCP) ou un ICMP port unreachable (UDP).
  - **sdrop :** (silent drop) paquet perdu mais pas de log.

L'IDS Snort

Les pré-processeurs :

Rôle des pré-processeurs

- **défragmentation** : "voir" plusieurs paquets à la fois
- reconstruction des flots, corrélation des évènements
- **décodage** pour pemettre l'analyse des **protocoles de haut niveau**
- ceci permet de **contrer certaines techniques d'insertion/évasion**

Pré-processeur frag3

- Reconstruit les paquets IP en tenant compte de l'architecture des hôtes (pour comprendre comment ils défragmentent).
- On peut aussi lui spécifier une TTL minimale.

Pré-processeur Stream5

- Il reconstruit des sessions TCP et UDP.
- Gère les données réenvoyées et les superpositions de données comme frag3

Techniques d'évasion/insertion

Insertion

Les techniques d'insertion consistent à insérer dans le flux des données :

- qui seront vues par l'IDS
- qui ne seront pas vues par la cible finale du flot de datagramme.

Le but est de noyer la signature de l'attaque et donc de tromper les reconnaissances par filtrage de motifs.

Mise en oeuvre des techniques d'insertion

Il s'agit de provoquer la perte du paquet entre l'IDS et la cible.

- Utilisation de TTL suffisamment faible pour s'arrêter au niveau de l'IDS (un traceroute permet de calculer la bonne valeur).
- Utilisation de combinaisons d'options IP qui vont faire refuser le paquet par la cible (par exemple en jouant avec les drapeaux réservés).

Techniques d'évasion/insertion

Evasion

Les techniques d'évasion consistent à transformer le contenu du flux de données qui seront vues par l'IDS, par des opérations qui seront comprises par la cible mais pas par l'IDS.

Mise en oeuvre des techniques d'évasion

Elles sont multiples et dépendantes des protocoles.

- Par exemple HTTP permet de remplacer un caractère par son code ascii précédé de '
- Le codage des caractères et en particulier UTF-8 permet de coder un caractère de plusieurs manières.
- Les octets nuls sont parfois interprétés de manière bizarre (ignorés silencieusement).

Analyse comportementale

IDS comportementaux

Le **principe** est d'avoir un modèle (par exemple statistique) du comportement "normal" et de détecter une **déviation par rapport au comportement normal**.

Apprentissage et détection

Nécessite deux phases :

- **phase d'apprentissage** : apprendre les situations normales (comportements habituels des utilisateurs, du trafic...) ; construire une base de connaissances caractérisant "ce qui est normal"
- **phase de détection** : analyse des mesures (fichier d'audit, sniffing réseau...) pour vérifier la conformité au modèle caractérisant un comportement "normal"

Rem : la base d'apprentissage doit être mise à jour régulièrement (non stationnarités, évolutions du comportement des utilisateurs, évolution du trafic...)

Analyse de trafic de niveau paquet/flot

Descripteurs Niveau paquet :

- Longueurs de paquets, temps inter-arrivée des paquets
- Différents champs d'entête (adresses et ports source et destination, protocole, ToS, drapeaux TCP ...)
- etc...

Descripteurs Niveau flot :

- Longueur du flot (nombre de paquets, quantité de données)
- Durée du flot
- 5 uplet : @IP Source et Destination, Port Source et Destination, Protocole

Les mesures de trafic de niveau flot réalisées par un point de capture sont remontées à un collecteur. NetFlow de CISCO, **Standard IPFIX**.

Analyse de trafic de niveau paquet/flot

Comptage en temps réel (et à haut débit)

- **Data stream mining :**
  - on ne peut pas découvrir d'anomalies dans un trafic très agrégé (ex : volumes de trafic par seconde sur une interface réseau)
  - Par contre une **observation du trafic à un grain très fin** (adresses IP, ports) révèle des anomalies très riches.
  - Exemple : paquets par IP destination par seconde $\Rightarrow$ détection de DoS ou scan de port.
  - Autres "anomalies" pouvant être détectées par comptage de certains types de paquets IP : scans de port, scans réseau, DoS, DDoS...
- **MAIS problème** : cardinalité de l'espace des données. @IP en IPv4 : 32 bits soit 2^{32} valeurs à surveiller (comptage des paquets/volumes de trafic par adresse IP).
- **SOLUTION algorithmique** : techniques de fouilles de données en temps réel (**data stream mining**)

Analyse de trafic de niveau paquet/flot

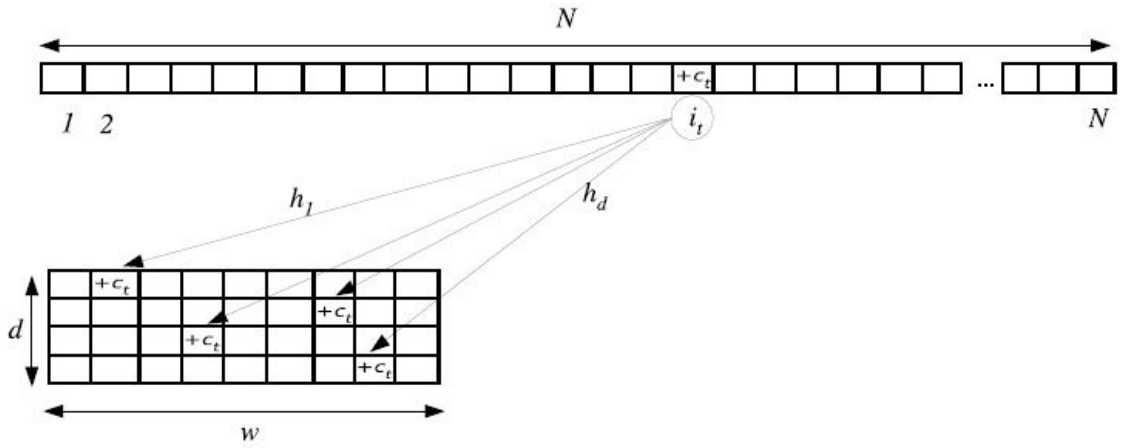
Algorithme Count Min Sketch (CMS)

- **flot de données** ; chaque élément du flot est caractérisé par un identifiant (ex : IP dest.) $i \in \{1, 2, \ldots, N\}$ (N très grand) et par un label $c \in \mathbb{R}$ (ex : taille du paquet en octets)
- on se donne d **fonctions de hachage** $h_1, h_2, \ldots, h_d$ de $\{1, 2, \ldots, N\}$ dans $\{0, 2, \ldots, w-1\}$ avec $w << N$; elles doivent avoir de bonnes propriétés de décorrélation entre elles
- on maintient **une table** S **appelée sketch** de taille d lignes et w colonnes ; cette table est une "mémoire" du trafic analysé (comptage de la quantité de trafic par identifiant $i \in \{1, 2, \ldots, N\}$)

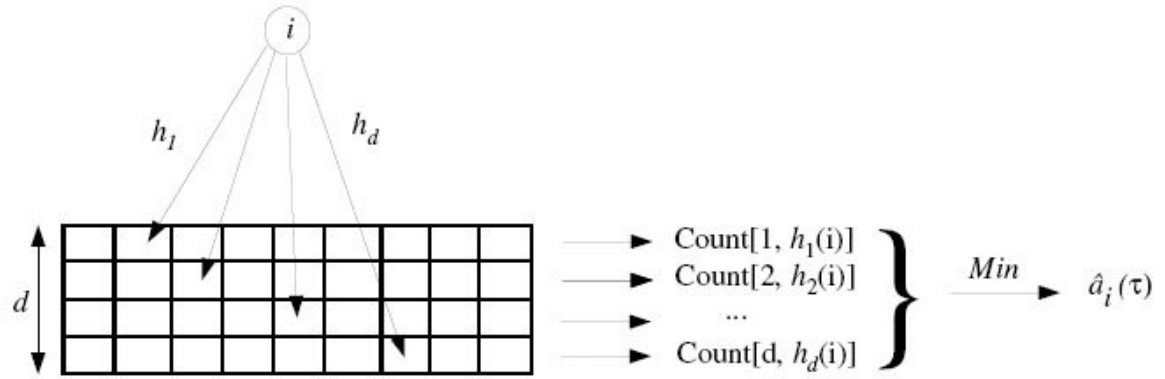
Analyse de trafic de niveau paquet/flot

Algorithme Count Min Sketch (CMS)

- **Mise à jour du sketch** : quand une nouvelle donnée (i_t, c_t) arrive on met à jour la table : $S[j, h_j(i_t)] += c_t, \forall j \in \{1, 2, \ldots, d\}$



- **Estimation** (du volume de trafic associé à un identifiant $i \in \{1, 2, ..., N\}$) : $\hat{A}(i) = \min_{j \in 1,...,d} S[j, h_j(i)]$



Analyse de trafic de niveau paquet/flot

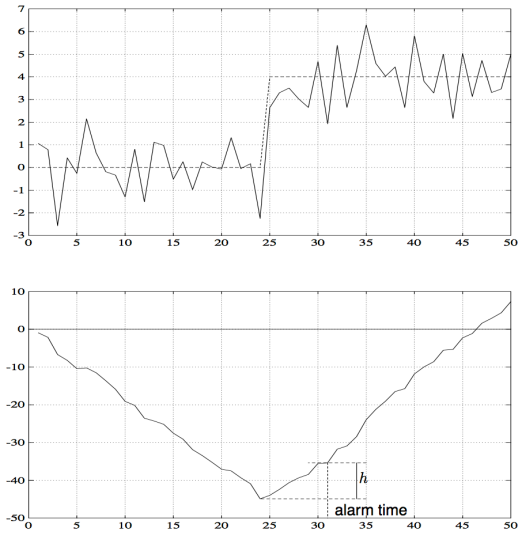
Détection d'un changement brusque : algorithme CUSUM (CUMulative SUM)

- théorie de la décision séquentielle ; détecter un instant de changement dans un flot de données séquentielles $x_1, x_2, x_3, ...$
- distribution avant (respectivement après) changement caractérisée par sa densité de probabilité f_1 (respectivement f_2) [modèles paramétriques]
- la "statistique" du test du CUSUM peut être maintenue de manière séquentielle : $S_{n+1} = max(0, S_n + log \frac{f_2(x_{n+1})}{f_1(x_{n+1})})$
- une alarme est levée lorsque la statistique S_n dépasse un certain seuil h ;
- Rem : on exploite le fait que $\mathbb{E}(log \frac{f_2(x_{n+1})}{f_1(x_{n+1})}) < 0$ avant changement et > 0 après changement
- influence du seuil h : plus h est élevé plus rares sont les fausses alarmes mais plus long est le délai de détection
- CUSUM : propriétés d'optimalités démontrées (minimise le délai de détection pour un taux de fausse alarme bornée)

Analyse de trafic de niveau paquet/flot

Détection d'un changement brusque : algorithme CUSUM (CUMulative SUM)

Exemple : détection d'un changement de moyenne dans une série temporelle avec le CUSUM



Analyse de trafic de niveau paquet/flot

Détection d'un changement brusque : algorithme CUSUM (CUMulative SUM)

Rem : versions non-paramétriques du CUSUM :

- Limitation : la distribution après changement est rarement connue
- Il existe des versions non paramétriques du CUSUM qui font très peu d'hypothèses sur les distributions avant et après changement.
- Exemple : NP-CUSUM

$$S_{n+1} = \min(0, S_n + x_n - (\mu + c \cdot \sigma))$$

avec μ et σ les moyenne et écart-type de la série temporelle avant changement. Cet algorithme est capable de détecter une augmentation de la moyenne de la série temporelle si son amplitude est supérieure à $c\sigma$.

MERCI POUR VOTRE ATTENTION !