Sécurité

Notions de base
Attaques OSI niveau 1-2

Carlos Aguilar

carlos.aguilar@enseeiht.fr

IRIT-IRT

Sources

Sources :

- Pierre-François Bonnefoi du Master CRYPTIS à Limoges
- Julien Cartigny du Master CRYPTIS à Limoges
- Dan Boneh de Stanford University
- Ron Rivest du M.I.T.
- Cédric Blancher (EADS)

Plan

Présentation

Découpage

- Une introduction rapide à plusieurs sujets indispensables
  - Concepts de la sécurité (confidentialité, intégrité, disponibilité, authentification, autorisation, accounting)
  - Outils non crypo (commutation, pare-feu, détection d'intrusion)
  - Outils crypto (chiffrement sym. et asym., fonctions de hachage, générateurs pseudo-aléatoires, signature, certification, authentification)
- Une description étendue des principales attaques réseau

Hors programme

Dénis de service, sécurisation des principales infrastructures de l'Internet, sécurité Wi-Fi, sécurité système et logicielle, gestion de la sécurité, sûreté de fonctionnement, etc.

Plan

Définitions (1/2)

Une première série

Propriétés associées à une information

Confidentialité :	ne pas être accessible aux personnes non-autorisées
Intégrité :	ne pas être modifiable par des personnes non-autorisées
Disponibilité :	être accessible aux personnes autorisées (très souvent associée à un service aussi)

Héritage de la sûreté de fonctionnement

Trois notions de la SdF (autres sont la fiabilité, maintenabilité) qui ont une importance certaine mais il n'y a pas que ça !

Définitions (2/2)

Une deuxième série

Cette fois c'est des mécanismes !

Authentification :	procédure permettant de vérifier une identité
Autorisation :	application d'une politique d'accès à des ressources
Traçabilité :	mise en place de logs permettant de savoir ce qui a été consommé ou fait
Auditabilité :	généralement post-mortem (analyse forensique), bénéficie grandement de la traçabilité

AAA(A) : Authentication, Authorisation and Accounting (and Auditing)

Factorisation proposée par Cisco

Achitecture réseau : contrôle des accès et enregistrement des usages faits

Menaces dans le cadre d'un réseau informatique

Objectifs classiques

- Écoute (cas simple ou élaboré)
- Contournement des moyens d'authentification/autorisation
- Déni de service (attaque physique, attaque sur protocole, vers)

Types d'attaquant

Internes (contrôlant un ou plusieurs utilisateurs) ou externes
Principalement internes : 80% [Jim Carr, Thwarting Insider Attacks]

Et hors du réseau ?

Tout change (objectifs, moyens) d'un contexte à l'autre (sécurité du logiciel, sécurité système, sécurité des cartes à puce, etc.)

Plan

Commutation et VLANs

Concentrateur (hub) vs Commutateur (switch)

Hub : chaque paquet reçu dans un port est diffusé dans tous les autres
Switch : ce n'est vrai que quand il ne connaît pas le MAC destination
... ou quand il n'arrive plus à marcher correctement ...

⇒ Utiliser des switchs
⇒ Utiliser la fonction port security
⇒ Sécuriser les locaux des équipements réseau

VLANs

Isolation au niveau 2 (comme s'il y avait plusieurs réseaux locaux physiquement séparés) par plages de ports/MACs/IPs

PVLANs

Par définition : VLAN de taille 1 défini au niveau 1 (notion d'isolated ports)
Exemple : hotel, aeroport, etc. (pas de comm. niveau 2 entre les utilisateurs)

Firewalls : principe

Tout ou rien

Le VLAN défini par plages IP :

- Même VLAN : on communique au niveau 2
- VLAN différent : on envoie au routeur qui décide de faire le pont ou pas

Principe du Firewall

Connecter deux réseaux au niveau 3 avec une politique plus fine que du tout ou rien
Filtrer en fonction :

- du protocole (TCP/UDP/ICMP)
- des ports et IPs origine et destination
- du sens de la connexion

Généralement on sépare la politique entrante et sortante

Firewalls : variantes

Firewalls de type routeur

Firewall sans états (stateless) :

- Examine chaque paquet indépendamment
- Peut bloquer les paquets avec le flag SYN ...

Firewall à etats (statefull) :

- Peut prendre en compte les paquets envoyés dans le passé
- Gestion de Notion de “transmission” UDP (indisp. pour le DNS)
- ICMP, fragmentation, port knocking ...

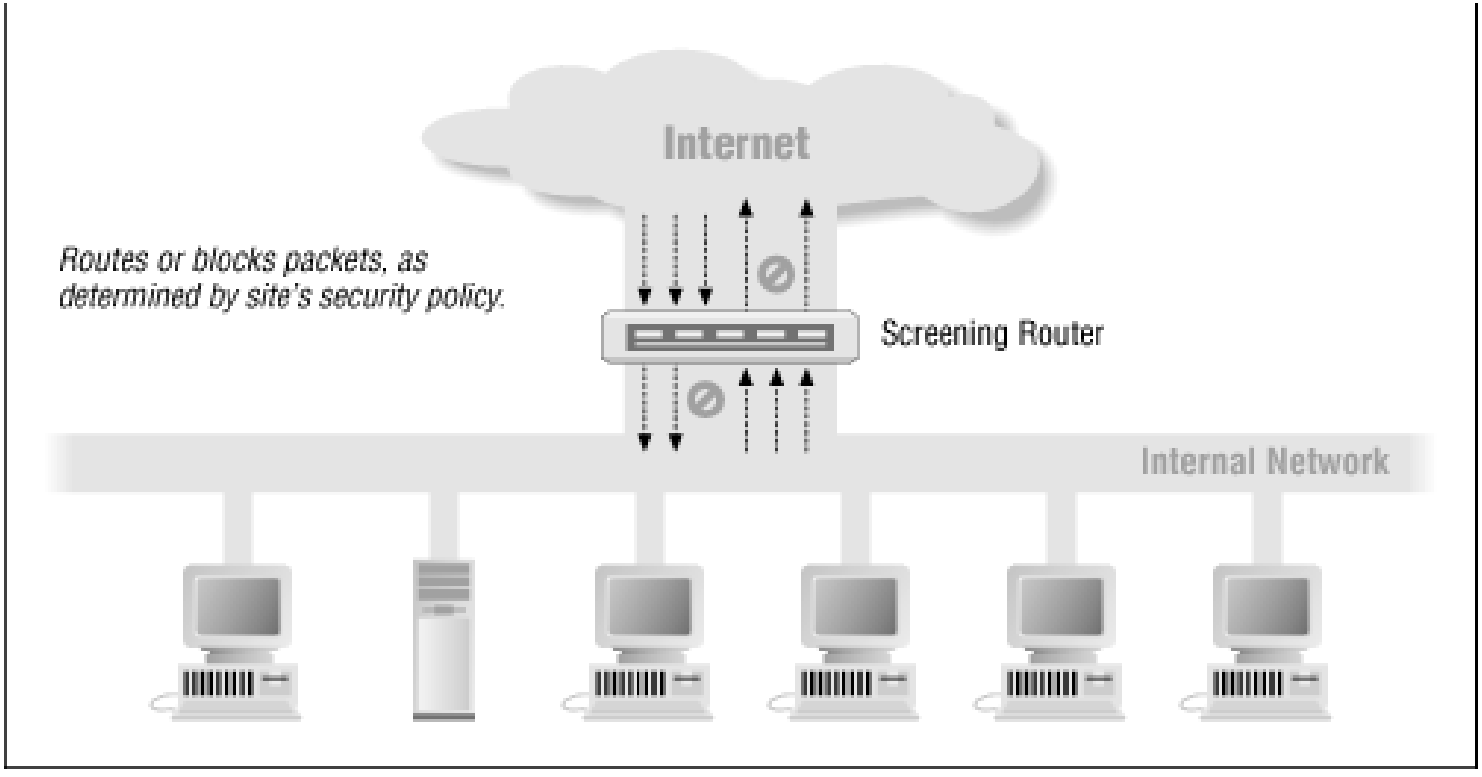
Firewall Applicatif :

- Protocoles dont les ports sont pas facilement prévisibles
- Vérification du protocole (pour éviter le contournement)

Firewall Personnel

Généralement statefull. ACL en fonction des applications !

Firewalls : architecture de type *screening router*



Firewalls : problèmes

Authentification basée sur l'IP

Les règles sont définies pour des machines
Contrôle de quelle machine par IP
Lien implicite machine-utilisateur

Tunnels et encapsulation

Tunnels SSH
HTTP CONNECT

Pas de défense en profondeur

Problème principal : les paquets sont routés !
$\rightarrow$ simplification des attaques si une faille est trouvée

Utilisation de relais applicatifs

Intérêt

Authentification des utilisateurs

- Peut utiliser une authentification forte (contrairement à l'IP)
- Mise en place de logs et politique en fonction de l'utilisateur

Mise en place d'une politique de sécurité par protocole

- Scan viral pour SMTP
- Pas de CONNECT en HTTP

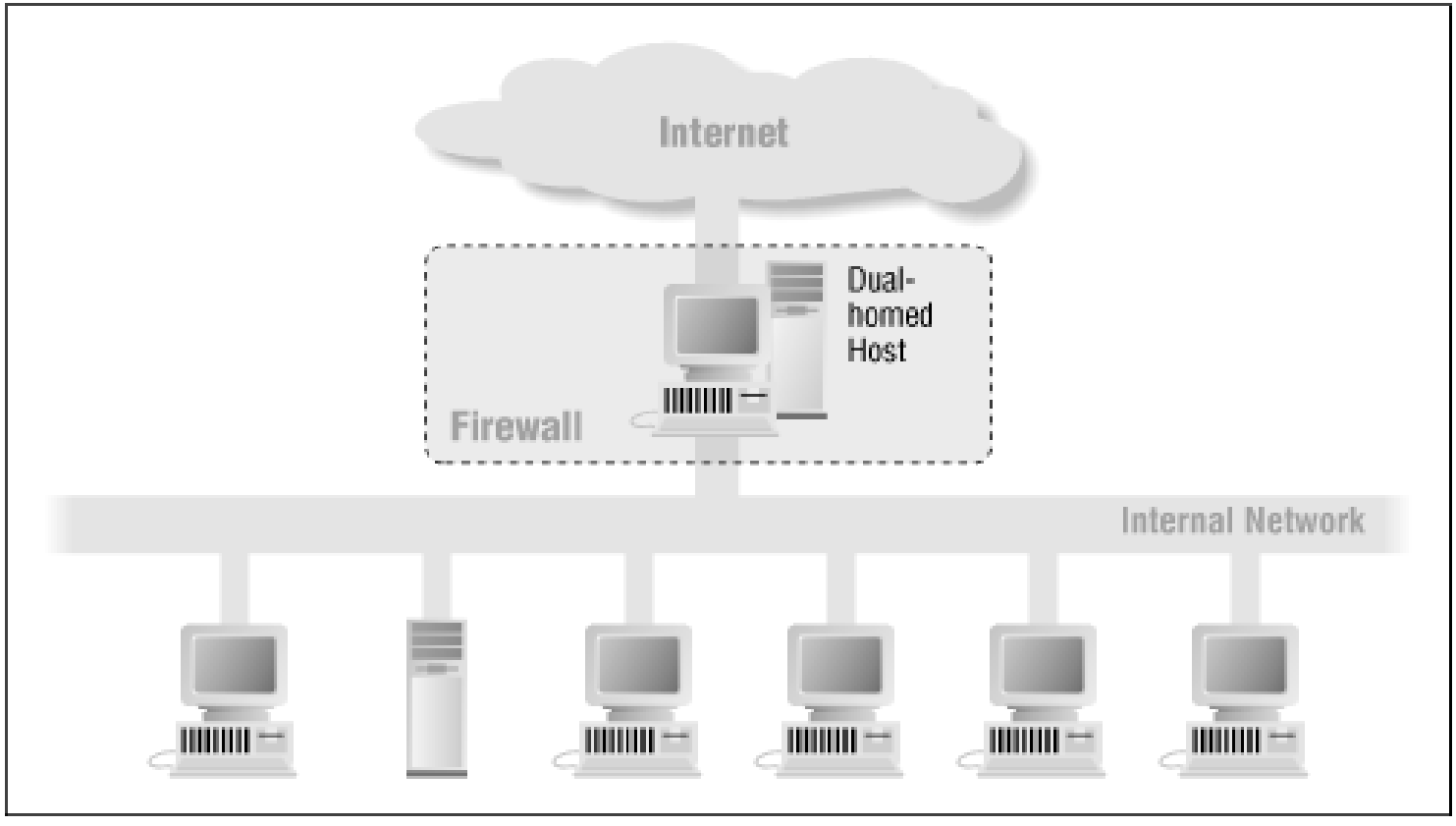
Paquets traités : il n'y a que les informations contenues qui passent !

Danger

Plus un programme analyse des données et est complexe plus il est facile de l'attaquer

- Réduire les services au minimum
- Audit du bastion/relais
- Prendre en compte qu'il peut être compromis

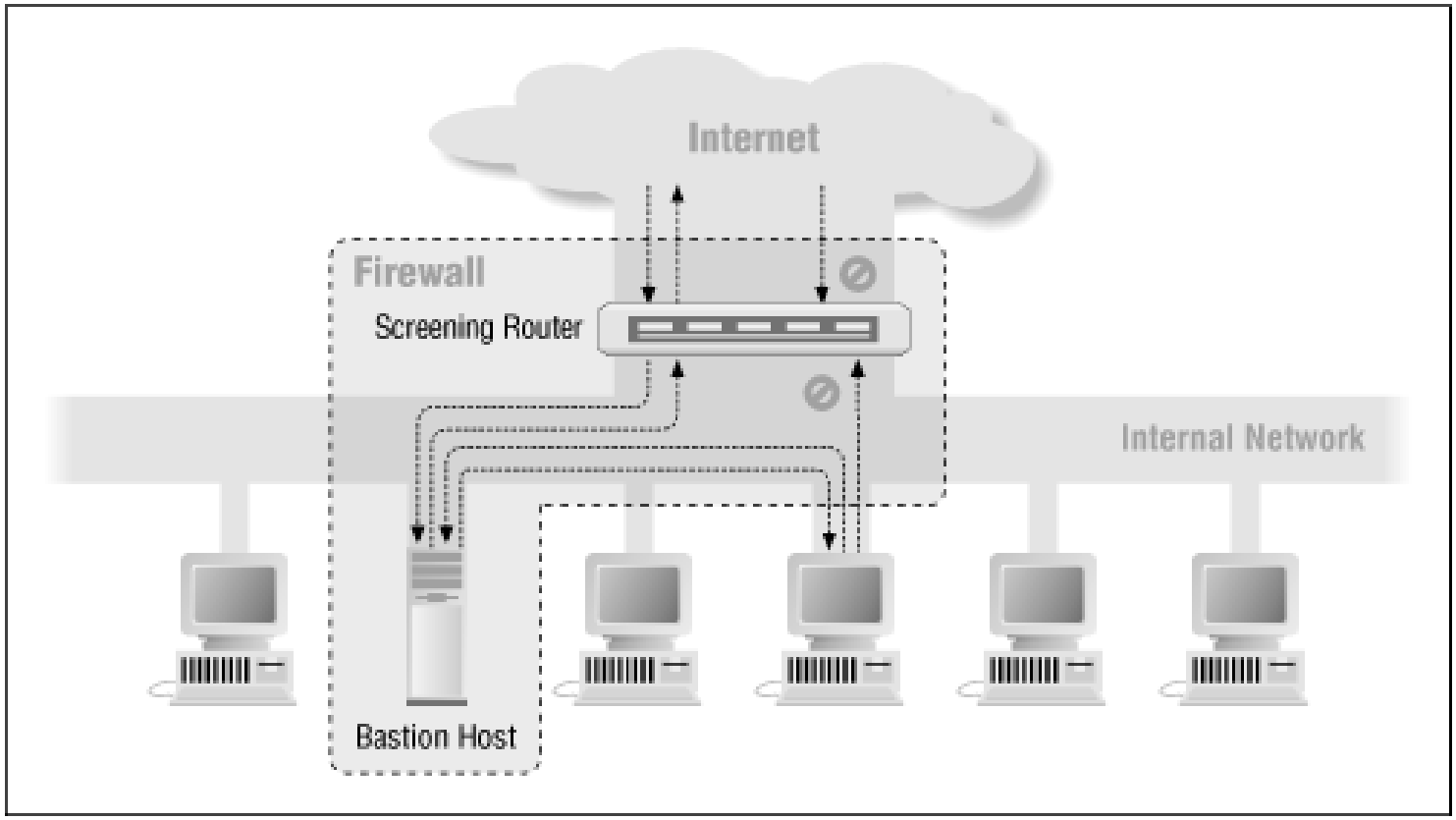
Architecture de type bastion relais



Le bastion ne route pas les paquets !

Mise en place de proxys entrants/sortants pour chaque service

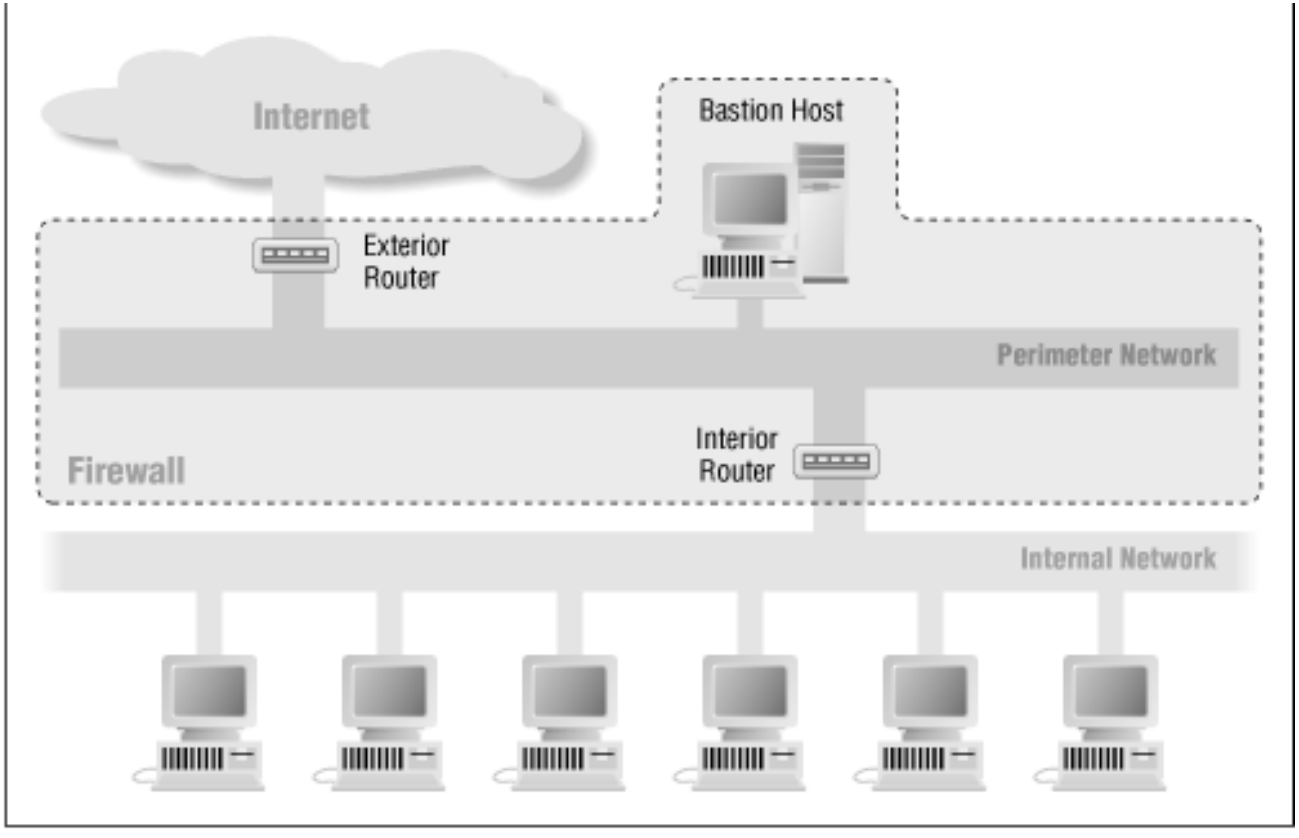
Architecture de type screening router + bastion



Avantages

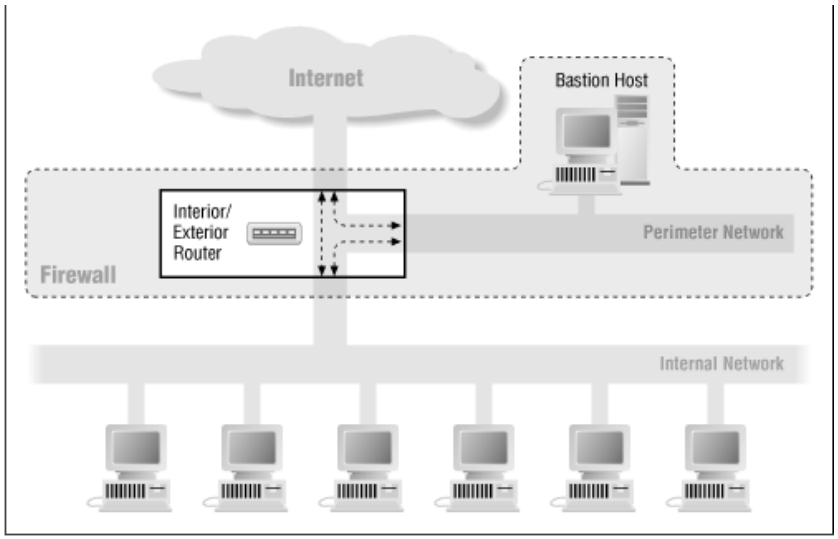
Protège le bastion
Plus modulable (en particulier pour les flux sortants)

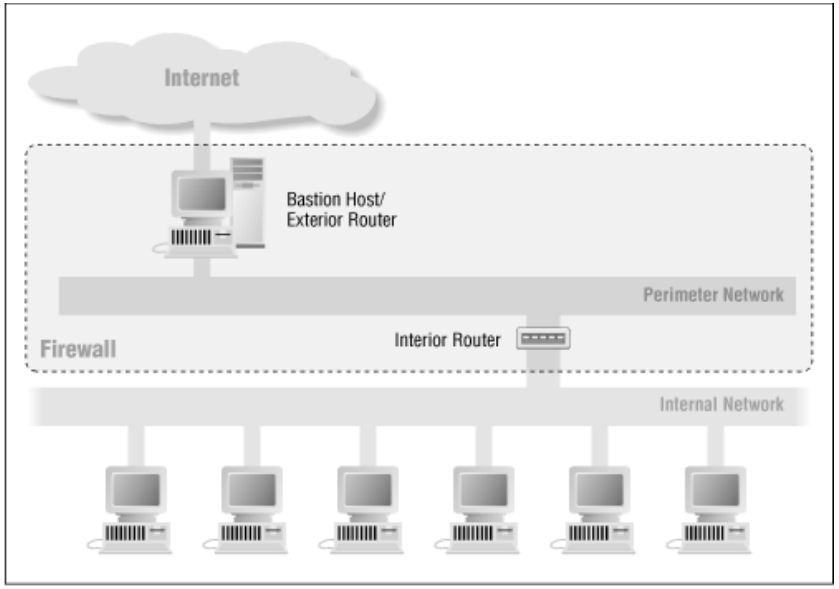
Architecture de type DMZ



Si le bastion tombe $\rightarrow$ architecture de type screening router
Plusieurs serveurs peuvent exister dans la DMZ (serveur web, SMTP, etc.)
Deuxième niveau de DMZ contenant des service (e.g. serveur SQL)

Architecture de type DMZ (variantes)





Moins de boulot

Si le bastion tombe $\rightarrow$ architecture de type screening router ...

Pas vrai si on fusionne le bastion et le routeur interne !

Plan

Systèmes de détection d'intrusion

Des outils par milliers

Host-based, network-based
Certains ont la notion de serveur et de sondes de détection

Deux principes de détection

Détection par signature (SNORT)

- Maintenir une base de données d'attaques connues
- Vérifier si l'activité correspond à la base

Détection d'anomalies (MCPAD, Cfengine)

- Apprendre quel sont les actions "normales" dans un réseau
- Lever des alertes quand il y qqch d'anormal

Problèmes

Trop de faux positifs, résistance aux nouvelles attaques

Plan

Attaques sur le lien physique : Variantes

Déni de service

Très simple (attention aux canaux) mais avec un effet local
Émettre en continu (téléphone, Ethernet, wifi)
Utilise généralement un appareil dédié (brouilleur)
Interdit par la loi (comme la plupart des autres attaques)

Écoute

En cablé : dispositif enregistreur dédié
En aérien :

- Communications aériennes
- Par rayonnement parasite (paranoïa ?)

Attaques sur le lien physique : Sniffers logiciels

Comment faire ?

Carte réseau en mode promiscuous (filtre MAC matériel désactivé)
`ifconfig eth0 promisc`
Capture des paquets (libpcap, tcpdump, sniffit, dsniff, ettercap, wireshark)

Contre-mesures

Segmentation et Chiffrement
Détection (anti-sniff, sniffdet)

- Génération de requêtes DNS (nombreuses/pour fake IPs)
- Réponses ARP pour des paquets avec en-tête Ethernet mal-formée
- Création d'entrées ARP fausses
- Temps de réponse en cas de charge réseau

$\rightarrow$ Anti Anti-sniff

MAC spoofing (1/2)

Méthode What/When/Where/Why/How !

What ?

Donner une fausse adresse MAC, généralement usurpée

Why ?

Contournement des moyens d'authentification/autorisation

- accès à une adresse IP dans un VLAN donné
- accès à une connexion internet (portail captif)

Écoute

- MAC Flooding (et donc écoute)

MAC spoofing (2/2)

How ?

```
[root]# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 00:A0:C9:29:3C:68
inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0
UP BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:4 dropped:0 overruns:0 carrier:4
collisions:0 txqueuelen:100
RX bytes:0 (0.0 b) TX bytes:168 (168.0 b)
Interrupt:11 Base address:0xdf00 Memory:df9ff000-df9ff038

[root]# ifconfig eth0 down
[root]# ifconfig eth0 hw ether 01:02:03:04:05:06
[root]# ifconfig eth0 up
[root]# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 01:02:03:04:05:06
inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0
UP BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:4 dropped:0 overruns:0 carrier:4
collisions:0 txqueuelen:100
RX bytes:0 (0.0 b) TX bytes:168 (168.0 b)
Interrupt:11 Base address:0xdf00 Memory:df9ff000-df9ff038
```

Remarque : attaques manifestes

Modèles d'attaquant

Un attaquant peut être :

- Interne/Externe
- Contrôler une machine / plusieures
- Passif/Actif

Attaquants actifs

Peuvent dévier des protocoles mais tout en voulant rester indétectables !

Attaques physiques

Écoute : indétectables
DoS : Clairement manifestes mais difficiles de localiser (surtout en aérien)

MAC spoofing

Ça dépend de ce qu'on fait (inventer, remplacer, cloner)

ARP spoofing/poisoning (1/5)

What ?

Pervertir le cache ARP d'une machine

Why ?

Écoute et Contournement de l'authentification/autorisation
Par détournement du trafic : associer notre MAC à l'IP de qqun d'autre

Background sur ARP : généralités

- Traduction liaison/réseau (datagramme IP $\rightarrow$ trame Ethernet)
- Chaque requête est lue par tous les ordinateurs du sous-réseau
- Utilisation de caches de traduction
- Point d'entrée : m.à.j. du cache lors de la réception d'une requête ARP

ARP spoofing/poisoning (2/5)

Background sur ARP : paquets

Deux types de paquet : requête (who-has) et réponse (is at). Contiennent :

1. Adresse physique de l'émetteur
2. Adresse IP de l'émetteur
3. Adresse physique du destinataire
4. Adresse IP du destinataire

Paquet générique pouvant être utilisé entre différentes couches 2 et 3
Une entête du protocole utilisé en niveau 2 (Ethernet) est ajoutée

Background sur ARP : utilisation

Quand on doit contacter une adresse IP dont on a pas le MAC on envoie une requête who-has [1||2||FF :FF :...||4] (+diffusion sur l'en-tête Ethernet)

Quand on envoie un is at on remplit tout normalement, la réponse est en 1

Background sur ARP : actualisation du cache

Quand on reçoit un who-has (qu'on traite) on note le lien 1-2
Quand on reçoit un is at (qu'on a demandé) on note le lien 1-2

ARP spoofing/poisoning (3/5)

Et un MAC Spoofing ?

Envoyer un trame Ethernet avec pour source l'adresse de notre victime ⇒ Modification du *Content Adressable Memory* du commutateur/routeur.

```
# Avant

Port | Adresse MAC
-------------------------
1    | 52:54:05:F4:62:30                    # alice
2    | 52:54:05:FD:DE:E5                    # bob
3    | 00:10:A4:9B:6D:81                    # charly

#Après

Port | Adresse MAC
-------------------------
1    | 52:54:05:F4:62:30                    # alice
2    |
3    | 00:10:A4:9B:6D:81; 52:54:05:FD:DE:E5 # bob, charly
```

Pbs : Manifeste, pbs de comm avec Bob, trames contradictoires...
Peut avoir un intérêt pour faire passer le commutateur en mode pont

ARP spoofing/poisoning (4/5)

Contexte

Charly veut qu'Alice ajoute une en-tête Ethernet avec l'adresse MAC de Charly aux paquets IP destinés à Bob

How ? Création

Répondre à la requête who-has avant bob ? bof ...
Envoyer un is at non demandé ? marche pas généralement

Utiliser la mise en cache lors de la réception d'un who-has

Envoyer à Alice un who-has $[MAC_c || IP_b || FF:FF:... || IP_a]$
Rq : Il est possible de l'envoyer en unicast ! (pas de contrôle de cohérence ARP-Ethernet)

ARP spoofing/poisoning (5/5)

How ? Mise à jour

Envoyer des is at régulièrement avec l'adresse MAC de Charly et IP de Bob

How ? Pratique

Utilisation de Scapy

```
Welcome to Scapy (2.1.0)
>>> h=ARP()
>>> h.show()
###[ ARP ]###
  hwtype= 0x1
  ptype= 0x800
  hwlen= 6
  plen= 4
  op= who-has
  hwsrc= 00:21:5d:c5:9e:5e
  psrc= 192.168.1.65
  hwdst= 00:00:00:00:00:00
  pdst= 0.0.0.0
```

```
>>> h.psrc=("192.168.1.254")
>>> h.pdst=("192.168.1.73")
>>> h.show()
###[ ARP ]###
  hwtype= 0x1
  ptype= 0x800
  hwlen= 6
  plen= 4
  op= who-has
  hwsrc= 00:21:5d:c5:9e:5e
  psrc= 192.168.1.254
  hwdst= 00:00:00:00:00:00
  pdst= 192.168.1.73
```

Contres

IDS (ARPWatch)
Commutateurs de niveau 3 / routeurs : Cache ARP Statique

Application aux attaques

Écoute

1. Charly réalise l'attaque décrite :
   - Il reçoit les paquets qu'Alice veut envoyer à Bob
2. Il active le routage sur sa machine et le configure pour renvoyer les données reçues à Bob après interception

Man In The Middle

1. Charly réalise l'attaque décrite à la fois sur Bob et sur Alice
   - Il reçoit les paquets qu'Alice veut envoyer à Bob
   - Il reçoit les paquets qu'Bob veut envoyer à Alice
2. Il active le routage dans les deux sens

DoS

Il suffit de pas router. Un DoS discret ! Le cas des DNS est particulièrement intéressant (en cas de corruption du DNS secondaire).

Fin !

Prochain cours

Cryptographie ?
Attaques sur IP