

---

# (In)Sécurité des réseaux locaux Ethernet

---

Cédric Blancher <blancher@cartel-securite.fr>

—

21 mai 2003

- Introduction
  - Présentation du contexte Ethernet

- Attaques de niveau 2/3
  - Filaire, wireless, équipements

- Corruption de cache ARP
  - Détail de l'attaque

## Ethernet comme protocole de niveau 1

- ▶ Fonctionne sur CSMA/CD
- ▶ Différents média : coaxial, paire torsadée, fibre
- ▶ Le signal est envoyé à tous
- ▶ Réseau de niveau 1 => répéteur (hub)
- ➔ On ne s'intéressera qu'aux bus en étoile

## Ethernet comme protocole de niveau 2

- ▶ Protocole de niveau 2
- ▶ Trame ethernet



Ethernet frame

- ▶ Adressage de niveau 2 sur 48 bits
- ▶ Réseau de niveau 2 => commutateurs (switches)

## Ethernet sans fil (WiFi)

- ▶ Fonctionne sur CSMA/CA
- ▶ Couche MAC dédiée basée sur un adressage ethernet
- ▶ Sécurité (!?) : authentification et chiffrement (WEP)
- ▶ Couche d'abstraction de type Ethernet
- ▶ Réseau WiFi => points d'accès

## Attaques possibles

- Niveau 1 : signal
- Niveau 2 : protocole, équipements
- Au dessus : gestion des couches

- Introduction

Présentation du contexte Ethernet

- Attaques de niveau 2/3

Filaire, wireless, équipements

- Corruption de cache ARP

Détail de l'attaque

## Ethernet filaire : niveau 1

- Écoute du signal
- Possible sur des répéteurs
- Attaque triviale
- ➔ Écoute
- ➔ Contexte trop “verbeux” pour aller plus loin

## Ethernet filaire : niveau 2

- ▶ Attaque protocolaire
- ▶ Usurpation d'adresse MAC
- ➔ Résultat aléatoire dépendant de l'équipement et de sa configuration
- ➔ Détournement partiel de flux, DoS

## Ethernet filaire : niveau 2

- Attaques des équipements et de leur configuration
  - Flood de table CAM (*macof*<sup>a</sup>) basé sur de l'usurpation
  - Saut de VLAN (*Scapy*<sup>b</sup>)
  - Saut de PVLAN (reroutage)
- ➔ Résultat très aléatoire souvent défavorable
- ➔ Écoute, accès non autorisés, DoS

---

<sup>a</sup><http://www.monkey.org/~dugsong/dsniff/>

<sup>b</sup><http://www.cartel-info.fr/pbiondi/scapy.html>

## Ethernet sans-fil : niveau 1

- ▶ Écoute du signal (mode moniteur)
- ▶ Brouillage
- ➔ Écoute, DoS

## Ethernet sans-fil : niveau 2

- ▶ Altération de la signalisation (*AirJack*<sup>a</sup>)
- ▶ Exploitation du roaming (*AirJack*, *FakeAP*<sup>b</sup>)
- ▶ Association
- ▶ Attaque du WEP (*AirSnort*<sup>c</sup>)
- ➔ Très efficace, difficile à contrer
- ➔ Détournement de flux (écoute/altération/injection) de flux, DoS

---

<sup>a</sup><http://802.11ninja.net/>

<sup>b</sup><http://www.blackalchemy.to/project/fakeap/>

<sup>c</sup><http://airsnort.shmoo.com/>

## Reconfiguration des équipements

- ▶ Interface de configuration
- ▶ SNMP
- ▶ Failles des OS (Cf. Phenoelit<sup>a</sup>)
- ▶ Protocoles propriétaires (Cf. Securite.org<sup>b</sup>)
- ➡ Fortement dépendant de la configuration
- ➡ Très efficace quand ça passe ;)

---

<sup>a</sup><http://www.phenoelit.de/fr/tools.html>

<sup>b</sup><http://www.securite.org/presentations/secip/>

Qu'est-ce que nous avons ?

- ▶ DoS
- ▶ Écoute
- ▶ Détournement partiel
- ▶ Détournement complet => écoute, altération, injection
- ➔ Une méthode générique ?

- Introduction
  - Présentation du contexte Ethernet
- Attaques de niveau 2/3
  - Filaire, wireless, équipements
- Corruption de cache ARP
  - Détail de l'attaque

## Lien Ethernet/IP : ARP

- ▶ Address Resolution Protocol - RFC 826
- ▶ Convertit une adresse IP en adresse MAC
- ➔ Indépendant des équipements de niveau 2
- ➔ Bon candidat pour une attaque !

## La trame ARP

0	7	15	23	31
ident. adresse physique		ident. adresse logique		
lg @ physique	lg @ logique	code		
adresse physique ...				
... de l'émetteur		adresse logique ...		
... de l'émetteur		adresse physique ...		
... du récepteur (inconnue)				
adresse logique du récepteur				

Comment ça marche ?

- ▶ Requête
  - who-has en broadcast
  - MAC/IP du demandeur
  - IP demandée
  
- ▶ Réponse
  - is-at en unicast
  - MAC/IP répondant
  - MAC/IP demandeur

## Exemple de trames

0x1		0x800
0x30	0x20	0x1
00:10:A4:9B:6D:81		
192.168.1.10		
00:00:00:00:00:00		
192.168.1.11		

### ARP request

0x1		0x800
0x30	0x20	0x2
00:04:76:40:65:5E		
192.168.1.11		
00:10:A4:9B:6D:81		
192.168.1.10		

### ARP reply

## Première idée : ARP Spoofing

- who-has émis en broadcast
  - répondre au lieu du vrai destinataire avec de fausses données
- ➔ Redirection de trafic
- ➔ Issue incertaine en fonction de qui répond en premier

## Le cache ARP

- ▶ Trop coûteux de faire une requête par émission de trame
- ▶ Introduction d'un cache
- ▶ Mécanisme opportuniste de gestion du cache : ajout/suppression/mise à jour
- ➔ Excellent candidat pour une attaque

Regardons d'un peu plus près...

- ▶ Créer une entrée
  - Sur réponse ou une requête
  - La RFC autorise les requêtes en unicast pour du keepalive
- ▶ Modifier une entrée
  - Sur réponse ou une requête
- ▶ Supprimer une entrée
  - Expiration de l'entrée ou cache plein
- ➔ Création facile mais peu intéressante
- ➔ Suppression inutile si on peut modifier

Paramètres sur lesquels on peut jouer :

- ▶ Ethernet : adresse MAC source
- ▶ Ethernet : adresse MAC destination
- ▶ ARP : source niv. 2
- ▶ ARP : destination niv. 2
- ▶ ARP : source niv. 3
- ▶ ARP : destination niv. 3

➔ Outil *arp-sk*<sup>a</sup>

---

<sup>a</sup><http://www.arp-sk.org>

## Principe de l'attaque

- ▶ Génération de réponses pour une autre IP
- ▶ Génération de requêtes pour une autre IP
- ▶ Génération d'ARP gratuit pour une autre IP
- ➔ On s'appuie sur la décorrélation des couches 2 et 3

## Exemple de trames corrompues

0x1		0x800
0x30	0x20	0x1
Spoofing MAC		
Spoofed IP		
00:00:00:00:00:00		
Target IP		

### Foiled ARP request

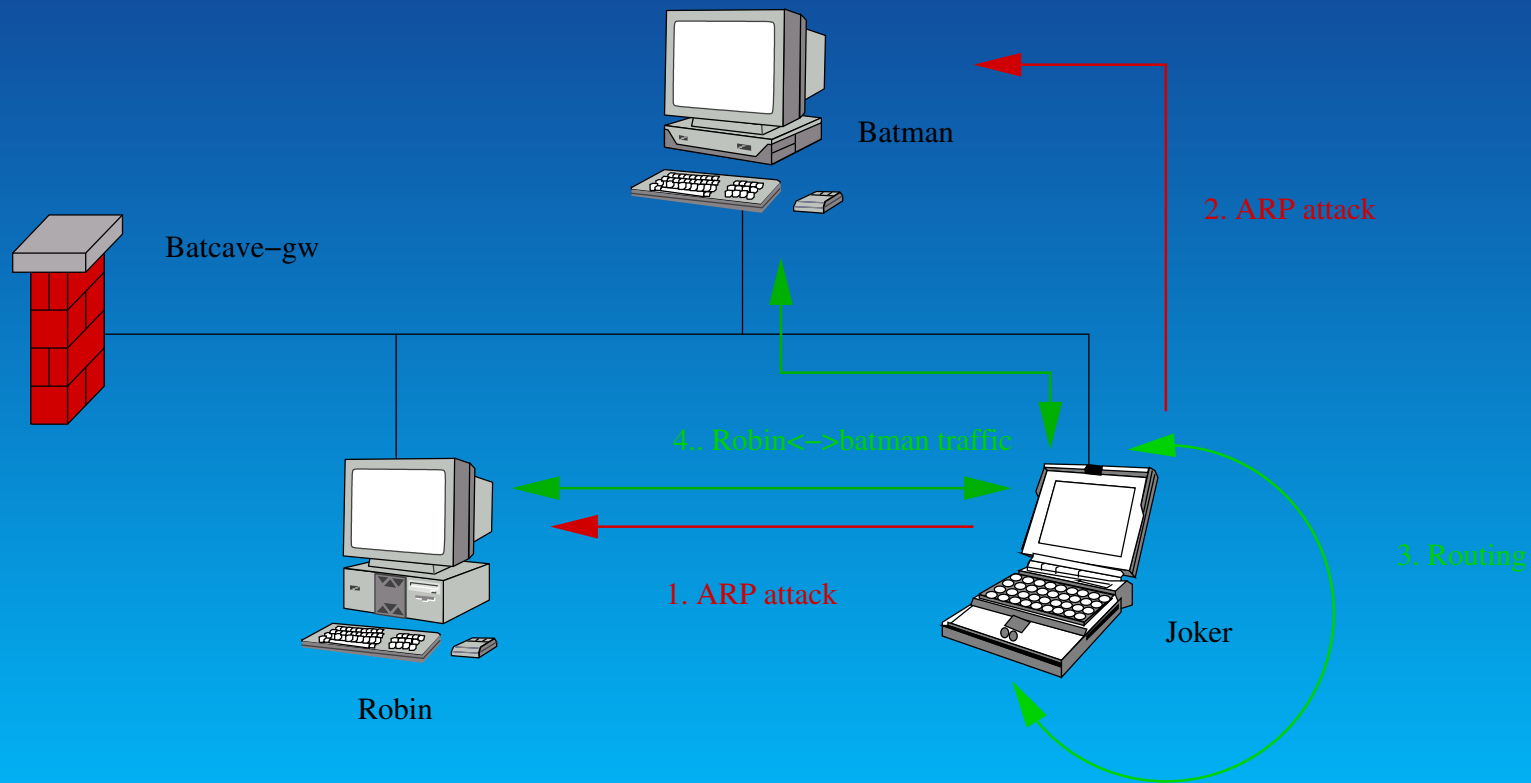
0x1		0x800
0x30	0x20	0x2
Spoofing MAC address		
Spoofed IP		
Target MAC address		
Target IP		

### Foiled ARP reply

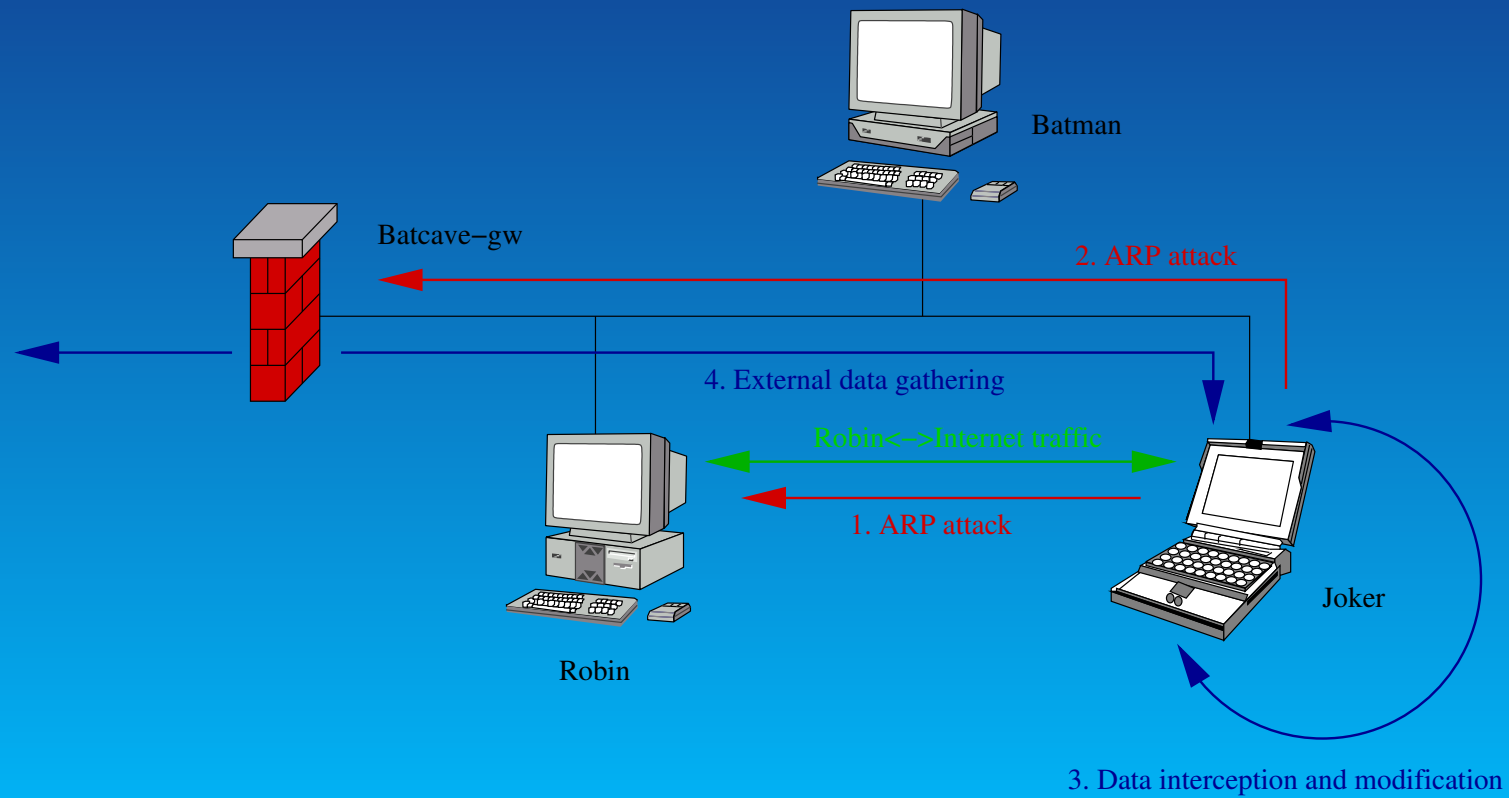
Les applications de la corruption de cache ARP...

- ▶ Écoute : on peut lire le contenu des flux détournés en mode normal
- ▶ Interception : on peut se placer comme proxy transparent
- ▶ Modification : on peut injecter des données dans les flux
- ▶ Vol : on peut prendre la place d'une des deux parties
- ▶ Déchiffrement : attaque MiM
- ▶ Usurpation : on peut aisément falsifier son IP
- ▶ DoS : destruction de flux réseau

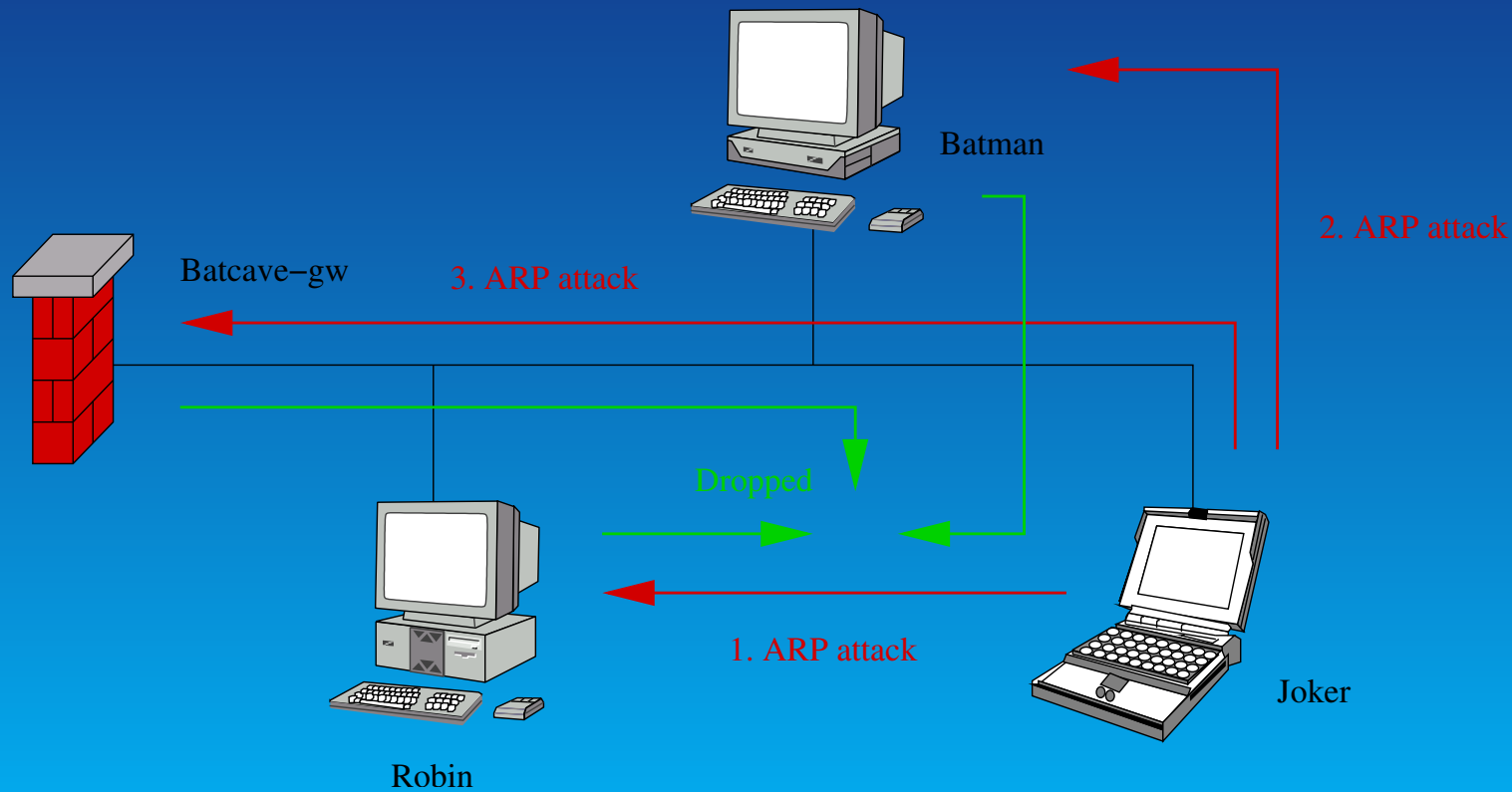
## MiM ARP pour écouter les flux



## Proxying transparent pour modifier et voler des flux

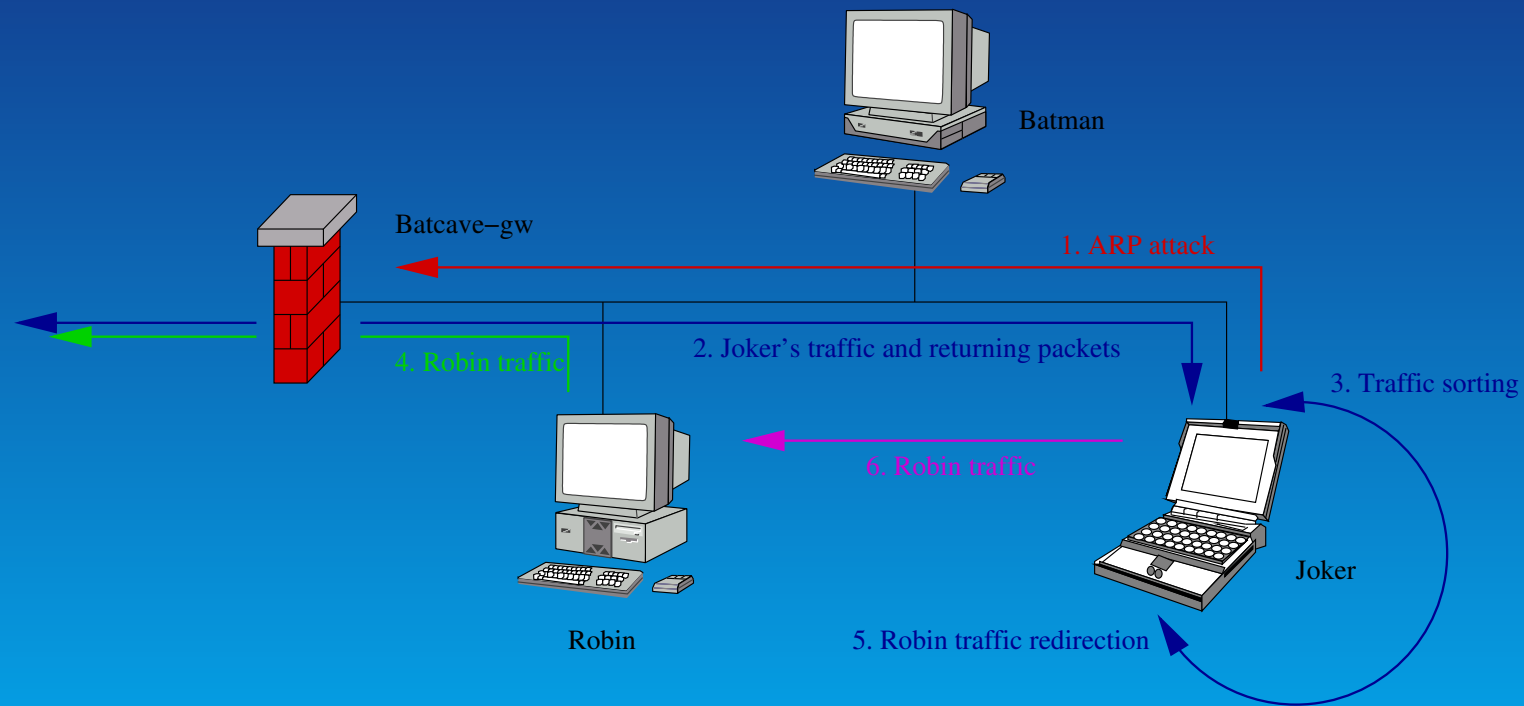


DoS



➡ Les machines attaquées vont vérifier leurs entrées...

“Smart IP spoofing<sup>a</sup>”



➔ On peut aussi faire du MiM ;)

<sup>a</sup><http://www.althes.fr/ressources/avis/smartspoofing.htm>

- ▶ Attaque possible sur tout support utilisant ARP
- ▶ Ethernet en particulier, filaire ou WiFi
- ➔ Une fois que l'attaquant est root, c'est tout le segment ethernet qui est perdu
- ➔ WiFi particulièrement sensible : on ne maîtrise pas l'accès au médium

- ▶ On a tendance à oublier les couches basses
- ▶ Pourtant très important dans l'élévation de privilège !
- ➡ En corrompant une couche, on corrompt tout ce qui passe au dessus

ARP est un protocole faible et facile à détourner : la sécurité n'était pas le but. On a besoin de quelque chose de plus solide pour *authentifier* les stations :

- ▶ Secure Link Layer<sup>a</sup>
- ▶ Sécurité niveau réseau (IPSEC)
- ▶ Sécurité applicative (SSH, SSL, etc.)

Il est clair que les switches ne sont pas des outils de sécurité

---

<sup>a</sup>[http://www.cs.wustl.edu/~fhunleth/projects/sll/sll\\_report.pdf](http://www.cs.wustl.edu/~fhunleth/projects/sll/sll_report.pdf)

Maîtriser l'accès au médium est primordial

- ▶ Sécurité physique pour Ethernet filaire
- ▶ Difficile pour le WiFi
- ➔ La bonne vieille serrure
- ➔ 802.1x, même si implémentations limitées en WiFi<sup>a</sup>

---

<sup>a</sup><http://www.cs.umd.edu/~waa/1x.pdf>

Pour se protéger efficacement ?

- ▶ Verrouillage maximale des équipements (Cf. Best Practices)
  - ▶ Cache ARP statique si disponible
  - ▶ Filtrage de niveau 2 ou de ARP (*ebtables*<sup>a</sup>)
  - ▶ Segmentation maximale
  - ▶ NIDS avec plugins spécialisés (*Prelude IDS*<sup>b</sup>)
- ➡ C'est lourd à mettre en œuvre...

---

<sup>a</sup><http://ebtables.sourceforge.net/>

<sup>b</sup><http://www.prelude-ids.org/>

<PUB>

**misc POWERED**  
Le magazine **100%** Sécurité Informatique

➔ MISC : magazine français, spécialisé en sécurité informatique<sup>a</sup>

</PUB>

---

<sup>a</sup><http://www.miscmag.com/>

- ➔ Présentation de Sean Convery (Cisco)<sup>a</sup>
- ➔ Site arp-sk<sup>b</sup>
- ➔ Article MISC3 "Jouer avec le protocole ARP"<sup>c</sup>

---

<sup>a</sup><http://www.arp-sk.org/doc/bh-us-02-convrey-switches.pdf>

<sup>b</sup><http://www.arp-sk.org>

<sup>c</sup><http://www.arp-sk.org/article/arp.html>

- ▶ DugSong pour le package *dsniff*
- ▶ Frédéric Raynal pour avoir implémenté *arp-sk* pour unix
- ▶ Éric Detoisien pour *winarp-sk* and *winarp-mim* sur Win32
- ▶ Laurent Licour et Vincent Royer pour leurs tests