

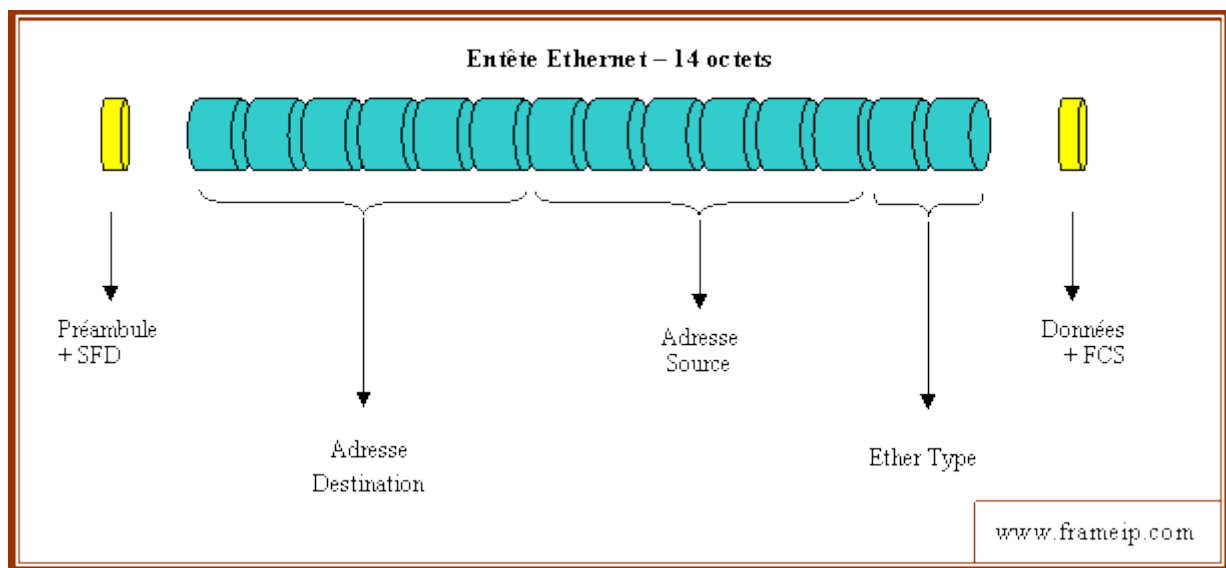
LES SNIFFERS ET ANTI-SNIFFERS

1 – Introduction aux sniffers et anti-sniffers

Les sniffers figurent parmi les premiers outils qui ont permis aux administrateurs systèmes d'analyser leurs réseaux et de localiser un problème avec précision. Ces mêmes outils sont aussi à la portée de hackers qui en usent également pour espionner les réseaux et voler tous genres de données. Ce chapitre définit ce qu'est un sniffer, explique son utilité, les risques qu'il présente, les services vulnérables au sniffing et enfin, il expose les sniffers les plus prisés. Nous détaillerons également dans ce chapitre les sniffers qui fonctionnent dans les réseaux partagés (principalement les réseaux utilisant des hubs) où les paquets envoyés à une destination seront aussi reçus par toutes les machines du réseau.

2 – Rappels

Entête d'une trame Ethernet :



Dans un réseau Ethernet partagé, toutes les machines sont reliées à un concentrateur réseau (hub). Un concentrateur amplifie le signal pour pouvoir le renvoyer vers toutes stations connectées. Toutes les trames Ethernet arrivant sur un hub sont donc renvoyées sur toutes les lignes, pour être reçues par toutes les stations. Dans le cas de réseaux importants par le nombre de stations connectées ou par l'importance du flux d'informations transférées, on ne peut utiliser des hubs. En effet, dès qu'une station émet quelque chose,

tout le monde l'entend et quand tout le monde commence à transmettre, les vitesses diminuent directement. Du point de vue sécuritaire, vu que toutes les trames passent par toutes les machines du réseau, un utilisateur malveillant situé sur le même réseau peut espionner un trafic qui ne lui est pas destiné. C'est le sniffing passif.

Les chapitres suivants détailleront ce type d'activité dans les réseaux Ethernet partagés. Chaque station connectée à un réseau Ethernet possède une adresse unique codée sur 48 bits (6 octets) appelée adresse Ethernet, ou adresse MAC (pour média access control). Cette adresse est située directement sur le coupleur Ethernet (généralement une carte d'interface réseau {network interface card, NIC) reliée au bus interne) et c'est pourquoi on l'appelle également adresse physique ou matérielle. Les trois premiers octets de l'adresse identifient le constructeur de la carte Ethernet tandis que les trois derniers représentent le numéro de série de cette carte. Donc, en principe, tous les coupleurs ont des adresses physiques différentes.

Dans un réseau Ethernet partagé, les données sont transmises dans les trames. Chaque trame contient plusieurs champs spéciaux, dont un correspondant à l'adresse MAC de l'émetteur et un autre à l'adresse MAC du destinataire. Lorsqu'une trame est envoyée sur le réseau, toutes les cartes réseau la reçoivent et la filtrent en comparant leur adresse MAC avec l'adresse du destinataire de la trame. Si les adresses MAC correspondent, la carte réseau transmet la trame pour traitement ; sinon, elle l'ignore. Chaque station traite donc uniquement les trames qui lui sont destinées, et cela permet d'éviter une surcharge de données. Lorsqu'une station envoie un message sur le réseau, elle peut l'envoyer à deux ou plusieurs destinataires, voire même à tout le réseau. Dans une trame Ethernet, l'adresse du destinataire doit appartenir à l'une des catégories suivantes :

- adresse monodestinataire (unicast address) : c'est l'adresse MAC d'une carte réseau. On utilise ce type d'adresse pour envoyer des données à une seule station sur le réseau ;
- adresse de diffusion multidestinataire (multicast address) : ce type d'adresse MAC est utilisé pour envoyer des données à un groupe de stations sur le réseau ;
- adresse de diffusion générale (broadcast address) : c'est une adresse réservée pour l'émission de données à toutes les stations du réseau. Elle est caractérisée par le fait que tous les bits sont à 1 (FF:FF:FF:FF:FF:FF en hexadécimal)

Le terme sniffer est plus populaire que des termes tels que « analyseur de protocole » et « analyseur de réseau ». Un sniffer est un programme qui permet de capturer tous les paquets circulant sur un réseau local (LAN) et qui permet d'afficher leurs contenus. Il peut capturer n'importe quelle information envoyée à travers un réseau local, et donc afficher aussi bien l'identité des utilisateurs que leurs mots de passe transmis par tout service transportant des données claires (non cryptées), tels que Telnet, DNS, SMTP, POP3, FTP et HTTP. Si les données ne sont pas cryptées et si elles passent par l'interface réseau de

la machine où s'exécute le sniffer, ce dernier les capture et les propose à la lecture directe. Mais c'est une arme à double tranchant. En effet, il est utilisé par l'administrateur réseau qui tente de résoudre les problèmes techniques de son entreprise, mais aussi par l'intrus qui cherche à espionner les données circulant dans un réseau local.

3 – Les Sniffers

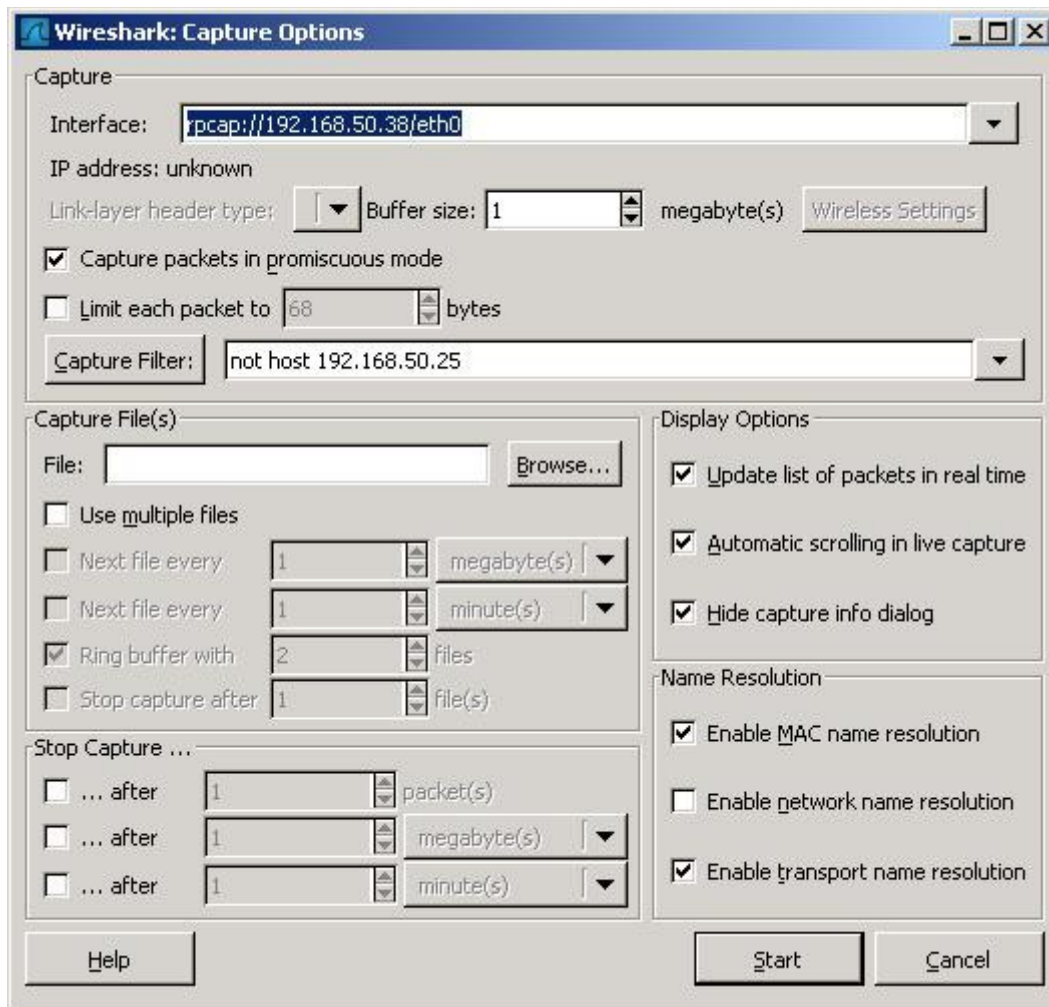
3.1 – Le mode Promiscuous

Les cartes réseau fonctionnent en deux modes, à savoir le mode normal et le mode promiscuous. Par défaut, les cartes réseau fonctionnent en mode normal. Ce mode permet à une carte de filtrer les paquets reçus sur l'adresse destination MAC. Ce type de filtrage est appelé filtrage matériel ou filtrage hardware. Par contre, le mode promiscuous consiste à accepter tous les paquets circulant dans un réseau, même ceux qui ne sont pas destinés à la carte. Dans ce cas, un sniffer collecte tout le trafic passant par cette carte. Dans des machines Unix, le mode promiscuous peut être activé en console grâce à la commande `# ifconfig promisc`. Si les machines « cible » et « pirate » sont sur le même réseau, le sniffer doit être lui aussi sur ce réseau. Si les machines sont sur deux réseaux différents, le sniffer doit être sur l'un des réseaux, et peut exécuter un « remote sniffer » pour lire les données à distance. Dans un réseau commuté (switch), le sniffing semble impossible à réaliser puisque seul le destinataire reçoit les trames qui lui sont envoyées. Nous verrons plus loin que certaines manipulations opérées sur le cache ARP permettent de faire du sniffing sur un réseau Ethernet, même s'il est commuté.

3.2 – Le remote Sniffing

Le remote sniffing, ou l'utilisation d'un sniffer à distance, permet d'obtenir les données circulant sur un autre réseau que celui sur lequel nous sommes. Considérons cet exemple : Un pirate est sur un réseau nommé A, mais voudrait sniffer le réseau voisin, appelé B. Admettons qu'il a pris possession d'une machine dans ce réseau B. Rappelons que nous sommes dans une configuration d'un réseau partagé. Le pirate va donc installer un client sniffer sur la machine du réseau B, qui va passivement récupérer et enregistrer toutes les données allant et venant sur ce réseau. Il enverra le tout à la machine du pirate dans le réseau A. Le réseau B qui en principe était impossible à sniffer est devenu donc très accessible. Le remote sniffing est toujours composé d'un client et d'un serveur, le client étant contrôlé par le serveur. Pour effectuer une telle attaque, il existe l'outil Rpcapd. Rpcapd est un démon (programme tournant en tâche de fond) qui capture le trafic sur une machine, et est capable d'envoyer les données récupérées à un sniffer comme ethereal qui facilite ainsi la lecture en différenciant les trames et les protocoles. Notons qu'il est utile d'exclure le trafic entre la machine local et la machine distante en utilisant les filtres d'ethereal.

Voici un exemple où nous excluons l'hôte 192.168.50.25 :



3.3 – Scénarios d'attaques

Nous allons montrer comment un utilisateur malveillant muni d'un sniffer peut espionner et collecter des informations confidentielles des utilisateurs d'un réseau. Pour cela, nous allons rapidement constater la facilité d'utilisation des sniffers pour espionner les autres utilisateurs du réseau cible et collecter des informations confidentielles. Des sniffers, tel que Sniffer Pro (Network Associates), ou Cain offrent des fonctionnalités avancées qui faciliteront la collecte d'informations confidentielles. Par exemple, il permet d'afficher les entêtes des paquets capturés des services réseaux, tels que FTP, HTTP, et SMTP. Ainsi, il suffit de lire les entêtes des paquets du service SMTP pour pouvoir lire le contenu des e-mails envoyés par un utilisateur dans le réseau. Tout service réseau envoyant des paquets non cryptés est vulnérable aux attaques de sniffing. Nous allons nous servir de deux sniffers bien connus pour leur efficacité : Cain et Ethereal.

Les cas d'attaques qui seront détaillés sont les suivants :

- Visualisation des connexions réseau ;
- Lecture des e-mails des utilisateurs ;
- Récupération des login et mots de passe des comptes e-mail, FTP, et authentification HTTP;
- Visualisation des sites web visités par les utilisateurs et lecture des e-mails;
- Réalisation des attaques de déni de service (DoS);
- Récupération d'une communication VoIP.

3.3.1 – Visualisation des connexions réseau

Même sans connaissance en réseau et protocoles TCP/IP, avec un sniffer, un utilisateur peut visualiser le trafic de son réseau notamment les différentes connexions. Ce type d'activité peut ennuyer les utilisateurs des réseaux puisqu'il touche à leur vie privée.

3.3.2 – Lecture des e-mails des utilisateurs

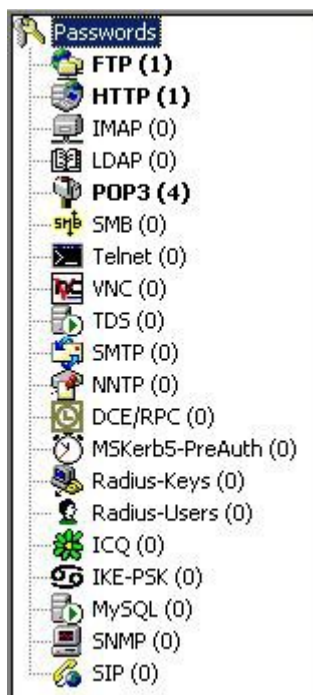
Un agresseur situé dans un réseau peut lire les e-mails envoyés par tous les autres utilisateurs connectés au segment du réseau où se trouve l'agresseur. Pour réaliser cette attaque, il suffit d'installer un sniffer permettant la capture des paquets selon des critères de filtrage définis par l'utilisateur. Dans notre cas, l'agresseur définit des filtres de sorte que le sniffer ne capture que les paquets dont le port source ou destination est égale à 25. En principe, tout trafic utilisant le port 25 est un trafic correspondant à des e-mails. Puisque les données dans les e-mails sont en général non cryptées, l'agresseur peut alors facilement lire le contenu des e-mails transitant par le réseau.

3.3.3 – Récupération des login et mots de passe

L'utilisateur d'un réseau peut frauduleusement visualiser les adresses des sites web visités par les autres utilisateurs du même réseau. Selon des critères de filtrage préalablement définis, la capture de paquets pourra être ciblée sur un port précis. Dans ce cas de figure, l'agresseur définit donc des filtres de telle sorte que le sniffer capture seulement les paquets dont le port source ou destination est égal à 80 (21 pour le FTP, etc...). En principe, tout trafic utilisant le port 80 ou 8080 est un trafic correspondant à des activités web (service HTTP). Il est clair que ce type d'attaque touche directement à la vie privée des utilisateurs réseaux. Il en va de même avec les services comme FTP, Telnet, ICQ, MSN et HTTP (le sniffer web_sniff est spécialisé dans l'écoute de requêtes HTTP). Il permet de connaître les adresses des pages visitées et d'afficher les mots de passe utilisés soit pour sortir sur Internet soit pour accéder à des informations sur un intranet.

Le service POP3 est utilisé lorsqu'un utilisateur veut télécharger ses e-mails à

partir d'un serveur de messagerie. Ce service envoie ces informations de login en clair. Dès que la victime se connecte du serveur POP3, le pirate récupère grâce au sniffer le login/mot de passe. Il en est de même avec la connexion à un serveur FTP, ou une simple page HTML qui demande une authentification (forums, site personnels etc). Les écrans ci-dessous montrent un exemple démontrant l'obtention du login et du mot de passe d'un utilisateur grâce au sniffer Cain :



Nous voyons ici la liste de protocoles que Cain peut sniffer, comme par exemple une connexion FTP avec le login/pass en clair :

Timestamp	FTP server	Client	Username	Password
18/10/2007 - 20:25:26	195.7.102.16	192.168.1.20	[REDACTED]	[REDACTED]

Et maintenant le login au forum d'Authsecu :

HTTP server	Client	Username	Password	URL
195.7.102.16	192.168.1.20	jeremya	[REDACTED]	http://www.authsecu.com/forum-securite/login.php

Et enfin la connexion à un serveur POP3 pour récupérer ses e-mails :

Timestamp	POP3 server	Client	Username	Password	AuthType
18/10/2007 - 20:23:15	80.12.242.8	192.168.1.20	[REDACTED]	[REDACTED]	ClearText
18/10/2007 - 20:25:17	80.12.242.8	192.168.1.20	[REDACTED]	[REDACTED]	ClearText
18/10/2007 - 20:25:20	80.12.242.8	192.168.1.20	[REDACTED]	[REDACTED]	ClearText
18/10/2007 - 20:25:23	195.7.102.4	192.168.1.20	[REDACTED]	[REDACTED]	ClearText

3.3.4 – Les attaques de déni de service (DoS)

Plusieurs sniffers tels que CommView intègrent des générateurs de paquets. Ces générateurs permettent de produire des paquets ARP, ICMP, TCP ou UDP. Il permet de mettre à jour les valeurs des champs des différents entêtes des paquets, et ensuite envoyer ces paquets à des machines cibles. On peut exploiter ces générateurs pour construire des paquets mal formés (par exemple, des paquets avec de faux champs et de fausses valeurs ou des paquets mal fragmentés) afin de causer des dénis de service au niveau des machines cibles.

Par exemple, l'outil frameip.exe permet de générer les trames de son choix

3.3.5 – Récupération d'une communication VoIP

La plupart des sniffers assez récent prennent désormais en charge le sniffing de communications dites VoIP (ToIP). Un utilisateur téléphonant grâce à son ordinateur par l'intermédiaire d'un logiciel comme Skype, VOIPBuster, MSN et tout autre protocole d'entreprise basé sur RTP peut être écouté par un tiers. La communication peut-être aisément capturée, comme nous le montrera l'exemple ci après avec le sniffer Cain. Les paquets sont récupérés puis analysés avant d'être enregistrés en format audible, le WAV. Attention, certains logiciels comme Skype chiffrent désormais les communications, ce qui rend l'analyse et l'enregistrement WAV beaucoup plus difficile, voire impossible pour un obtenir un résultat acceptable.




On remarque que l'on obtient bien un fichier WAV qui, en lançant grâce au menu contextuel permet d'écouter la conversion en cours.

Started	Closed	IP1 (Codec)	IP2 (Codec)	File
11/09/2007-20:11:39	18/10/2007-20:34:14	[REDACTED] 116.67:6902	[REDACTED] 192.168.1.20:11015	
11/09/2007-20:11:44	11/09/2007-20:11:43	[REDACTED] 192.168.1.20:11015	[REDACTED] 192.168.1.150:34018	RTP-200.wav

Play
Remove Delete
Remove All

Le cas ci-après montre que la communication n'a pas été enregistrée, le 'codec' n'étant pas reconnu, c'est-à-dire que le logiciel utilisé pour la communication

envoi des paquets cryptés, ce que ne reconnaît pas encore notre sniffer.

	11/09/2007-20:12:27	11/09/2007-20:12:27	192.168.1.20:11015	192.168.1.20:26584	codec unknown
	11/09/2007-20:12:31	11/09/2007-20:12:31	192.168.1.20:11015	192.168.1.20:67.191:64351	codec unknown
	11/09/2007-20:12:31	11/09/2007-20:12:31	192.168.1.20:11015	192.168.1.20:5.87.0:18268	codec not supported
	11/09/2007-20:12:31	11/09/2007-20:12:31	192.168.1.20:11015	192.168.1.20:2.204.105:6488	codec unknown

3.3.6 – Conclusion

Les exemples présentés sont réels et visent une meilleure sensibilisation aux problèmes relatifs au sniffing. Les divers scénarios exposés démontrent la facilité de réalisation des attaques avec des sniffers. Avec peu de connaissance réseau, un pirate peut aisément mener plusieurs de ces attaques. Ces scénarios démontrent également les faiblesses et les vulnérabilités liées aux protocoles et services réseaux, notamment le protocole ARP et les services Internet non cryptés. Il est donc impératif de disposer de techniques et d'outils efficaces pour détecter ces activités malveillantes.

Le chapitre suivant présente les techniques et les outils de détection des sniffers dans les réseaux partagés.

4 – Les Anti-Sniffers

La détection des cartes réseau en mode promiscuous permet de déceler la présence de toute activité suspecte visant à capturer ou à analyser le trafic.

4.1 – Le concept du mode promiscuous

Les cartes Ethernet réseau sont construites par défaut avec une adresse MAC unique stockée sur 6 octets. Chaque constructeur dispose de sa propre plage qui permet ainsi d'attribuer un identifiant unique pour chaque interface. Les adresses MAC sont obtenues grâce au protocole ARP et chaque résultat est stocké dans un cache local, de façon à éviter d'initier pour chaque nouveau paquet une nouvelle requête. Une interface réseau en mode normal accepte uniquement les paquets qui lui sont destinés ou bien les paquets de diffusion (broadcast) ou de multidestination (multicast). Cependant il est possible d'activer le mode promiscuous dans les interfaces réseaux pour récupérer de façon transparente l'ensemble des paquets circulant sur le réseau et destinés à des tiers. La détection d'une activité de ce genre est alors très difficile puisqu'il s'agit d'un comportement passif qui n'interagit pas avec le fonctionnement normal du réseau.

4.2 – Les filtres système

Avant d'atteindre le noyau d'un système d'exploitation pour traitement, un paquet passe en général par deux filtres, à savoir le filtre matériel (hardware) localisé au niveau de la carte réseau et le filtre logiciel au niveau du noyau du système d'exploitation. Ainsi, la carte réseau assure d'abord le filtrage matériel

des paquets ; ensuite, si le paquet est accepté par le filtre matériel, il est alors transmis vers le filtre logiciel qui le transmet à son tour au noyau du système pour traitement.

Le mode promiscuous permet essentiellement de capturer l'ensemble des paquets circulant sur un réseau. L'activation de ce mode entraîne la désactivation du seul filtre matériel, le filtre logiciel au niveau du noyau du système demeurant toutefois actif.

4.3 – Les filtres matériel

Le filtre matériel est localisé au niveau de l'interface réseau. Il a pour but de ne transmettre au noyau que les paquets destinés à la machine elle-même en tant qu'hôte unique (unicast) ou hôte d'un réseau (broadcast et multicast). Le filtre matériel dispose donc de plusieurs modes de fonctionnement dont voici la liste :

- unicast : considérant que chaque interface réseau reçoit une adresse MAC unique, le filtre matériel ne laisse donc passer que les paquets destinés à cette même adresse MAC ;
- broadcast : les paquets broadcast sont destinés à l'ensemble des hôtes du réseau et permettent généralement de véhiculer des messages de notification. Le protocole ARP qui permet de trouver l'adresse MAC d'un hôte à partir de son adresse IP est un exemple de protocole qui génère des requêtes ARP broadcast. Les paquets broadcast ARP utilisent l'adresse destination MAC FF:FF:FF:FF:FF:FF. Dans ce cas le filtre matériel autorise les paquets broadcast à atteindre le filtre logiciel du noyau du système ;
- multicast : un paquet multicast est destiné à un groupe de hôtes. L'utilisation d'adresse multicast permet d'adresser n machines à partir d'une source unique. Son mode de fonctionnement repose sur l'utilisation de groupe multicast. Ainsi toutes les machines d'un groupe recevront le paquet en question. L'utilisation de ce mode suppose l'attribution d'une adresse MAC multicast pour le groupe en question. Le filtre matériel autorisera donc dans ce cas les paquets multicast « valides » à atteindre le noyau du système. Une adresse multicast commence par les trois octets suivants: 01:00:5E:X:Y:Z ;
- tout multicast : ce mode, aussi appelé « all multicast », est une extension du mode multicast où tous les paquets multicast seront transmis au filtre logiciel du noyau du système d'exploitation ;
- promiscuous : une interface réseau en mode promiscuous transmet l'ensemble des paquets véhiculés sur le réseau vers le filtre logiciel du noyau du système sans même analyser l'adresse destination MAC.

D'une façon générale, les interfaces réseau activent par défaut, au niveau du filtre matériel, les modes suivants : unicast, broadcast et multicast d'adresse 1 (01:00:5E:00:00:01). L'interface réseau d'une machine X et à envoyer à partir d'une autre machine une requête ARP recherchant l'adresse MAC de la

machine d'adresse IP X. L'adresse destination MAC de la requête ARP est égale à une adresse quelconque, choisie arbitrairement. En principe, elle doit être une adresse broadcast. Alors qu'on s'attend à ce que l'interface réseau de X récupère le paquet et le transmette au noyau qui se chargera ensuite d'y répondre, la machine X ne génère aucune réponse ARP. C'est le filtre logiciel, localisé à la suite du filtre matériel, qui empêche la requête d'atteindre le noyau du système. Ce filtre effectue donc un deuxième niveau de filtrage dans la chaîne d'acheminement du paquet vers le noyau.

Le filtre logiciel est différent d'un système à un autre. Maîtriser son fonctionnement exige, dans le cas de Linux, où le code source est libre (open source), une lecture attentive des instructions contenues dans ce code. Dans Windows, ce code est non libre et demande, par conséquent, une analyse comportementale approfondie du système. Par exemple, l'analyse du code source de Linux montre que son filtre logiciel sépare les paquets en 4 catégories : broadcast, multicast, TOUS ou OTHERHOST. Les paquets broadcast sont relatifs à l'adresse MAC broadcast FF:FF:FF:FF:FF:FF alors les paquets multicast caractérisent l'ensemble des paquets avec le bit de groupe positionné (01 pour l'octet de poids fort : 01:X:Y:Z:V:W) TOUS caractérise l'ensemble des paquets ayant pour destination l'adresse MAC de l'interface réseau tandis que OTHERHOST définit l'ensemble des paquets ayant pour destination une adresse MAC différente.

4.4 – Les méthodes de détection des sniffers

Cette section est consacrée à la présentation des méthodes les plus connues de détection des sniffers dans un réseau Ethernet partagé.

4.4.1 – La méthode du DNS

Les sniffers ne sont pas totalement passifs. En exécutant automatiquement des requêtes DNS inverses {reverse DNS lookup} pour traduire les adresses IP des paquets capturés, la plupart génèrent du trafic sur le réseau. Le principe de la méthode du DNS consiste à exploiter cette fonctionnalité commune des sniffers. Ainsi, pour détecter un trafic est généré par un sniffer, il suffit de rechercher des requêtes DNS inverses et de les distinguer des vraies requêtes DNS inverses. Pour ce faire, un faux trafic est généré sur le segment du réseau Ethernet avec une fausse adresse destination IP non utilisable (adresse de test). En plus, ce faux trafic est généré sur le segment du réseau Ethernet avec une fausse adresse destination MAC. Ce faux trafic devrait être ignoré par toutes les machines en mode normal du réseau (filtrage par le filtre matériel). Si une requête DNS inverse correspondant à la fausse adresse de test est capturée, ceci prouverait que la machine est probablement en train d'exécuter un sniffer ; car seules les machines en mode promiscuous peuvent capturer ce genre de faux trafic. Cependant, cette méthode présente l'inconvénient d'être devenue très connue par les pirates, qui désactivent, désormais, la génération des requêtes DNS inverses dans de nouveaux sniffers comme Ethereal. Cette fonctionnalité reste quand même présente en option. Cette méthode reste tout de même utile pour détecter la majorité des intrusions car les pirates

téléchargent souvent des sniffers sans même maîtriser leur mode de fonctionnement.

Voici un exemple :

On suppose que les machines d'un réseau Ethernet ont des adresses IP appartenant à la plage 172.16.16.1 – 172.16.16.100. Un faux paquet ping ICMP est généré à l'adresse IP 172.16.16.182 non utilisable. Ce faux paquet est généré par une machine de test d'adresse 172.16.16.2. La machine d'adresse IP 172.16.16.20 est en mode promiscuous et exécute un sniffer. Les valeurs des principaux champs du faux paquet ping ICMP sont mentionnées dans le tableau :

Entête Ethernet

Adresse source MAC = Adresse MAC de la machine de test : 00:02:A5:B6:E3:82

Adresse destination MAC = Une fausse adresse MAC (par exemple : AA-AA-AA-AA-AA-AA)

Type Ethernet = 0x0800 (IP message)

Entête IP

Adresse source IP = Adresse IP de la machine de test : 172.16.16.2

Adresse destination IP = L'adresse IP non utilisable : 172.16.16.182

Entête ICMP

Type = 8 (echo request)

Code = 0

On peut utiliser l'outil frameip.exe pour générer et envoyer le faux paquet ICMP avec les valeurs des principaux champs.

Les machines en mode normal bloqueront le faux paquet ping ICMP. Cependant, les machines en mode promiscuous dont l'option « génération des requêtes DNS inverses » est activée, généreront des requêtes DNS inverses vers un serveur DNS pour rechercher le nom de la machine avec l'adresse IP 172.16.16.182. La machine de test d'adresse IP 172.16.16.2 exécute un sniffer pour visualiser le trafic et capturer particulièrement les requêtes DNS inverses.

La machine d'adresse IP 172.16.16.20 a envoyé une requête DNS inverse vers le serveur DNS d'adresse 193.95.66.10 à la recherche du nom de la machine d'adresse IP 172.16.16.182. Il est donc clair que la machine émettrice est en mode promiscuous, exécutant un sniffer qui génère des requêtes DNS inverses.

4.4.2 – La méthode pots de miel (honeypots)

Si la plupart des méthodes de détection des sniffers fonctionnent uniquement réseau local, la méthode pots de miel Honeypots, elle, fonctionne partout. Elle exploite le fait que nombreux protocoles transmettent des mots de passe non cryptés, et que les intrus cherchent souvent des mots de passe et des logins. Elle consiste simplement à installer un client et un serveur transmettant des paquets non cryptés (par exemple un serveur FTP ou HTTP) dans le réseau. L'accès au serveur exige de l'utilisateur de s'authentifier en fournissant un login et un mot de passe (accès non anonyme). Le client ouvre une session au

serveur en fournissant un login et un mot de passe valides. Le serveur qui est complètement virtuel, est configuré avec des comptes virtuels.

Une fois qu'un intrus a obtenu le login et le mot de passe dans le réseau, il veut accéder au serveur en utilisant ces informations. Un système de détection d'intrusion ou un sniffer peut être configuré pour noter ces occurrences, alertant qu'un intrus a trouvé le trafic et a essayé d'employer l'information.

Cette méthode est toutefois limitée puisque l'intrus, pour une raison ou pour une autre, peut retarder sa tentative d'accéder au serveur. De plus, il peut ne pas être intéressé par ce type de trafic. Il utilise des filtres sur les paquets et ainsi il ne voit pas les paquets relatifs à la connexion avec le serveur virtuel.

4.4.3 – La méthode de l'hôte local

Détecter localement un processus qui fait fonctionner le sniffer est difficile parce que le nom de ce processus peut être déguisé en quelque chose d'anodin. La seule manière pour détecter un sniffer dans ce cas, est de contrôler si la carte réseau n'est pas en mode promiscuous. Une machine ne devrait jamais être en mode promiscuous, sauf si son fonctionnement l'exige (cas pour un routeur ou un firewall). Le fait que la carte réseau soit en mode promiscuous est une forte indication qu'un sniffer est en cours exécution.

Sous la plate-forme Unix, il existe différentes applications qui vérifient si la carte réseau locale est en mode promiscuous, comme l'outil CPM (check promiscuous mode). Une autre méthode est d'exécuter la commande « ifconfig -a » qui énumérera les interfaces réseaux, et affichera toutes les informations à leur sujet. Le mot PROMISC signifie que la carte réseau est en mode promiscuous. Ainsi, la commande « ifconfig -a | grep PROMISC » permet d'afficher uniquement les cartes réseau en mode promiscuous. Notons que l'utilitaire ifconfig est parfois remplacé par des pirates pour éviter d'être découverts.

Les systèmes Irix et Solaris n'ont aucun flag d'indication sur le mode promiscuous. De même sous Windows, aucune commande ne permet de vérifier le mode promiscuous pour la machine locale.

La méthode de l'hôte local exige un accès physique à la machine cible pour identifier le mode de sa carte réseau. Ceci n'est pas pratique du tout car le but d'un administrateur réseau est de détecter à distance les machines du réseau avec des cartes en mode promiscuous.

Expérience sous Linux :

Nous allons essayer quatre commandes qui permettent de savoir si la carte réseau local est en mode promiscuous sous Linux. Nous allons également les comparer en vue de choisir les commandes les plus robustes. Ces commandes sont : ifconfig, CPM (check promiscuous mode), ip link list et ip address show. L'expérience se déroule en quatre étapes.

- 1ère étape, Tout d'abord, toutes les cartes réseau sont en mode normal. On exécute les deux commandes ifconfig et CPM. Les cartes réseau eth0 et lo sont en mode normal. Ensuite, on exécute les deux commandes ip

link list et ip address show. Les cartes réseau eth0 et lo sont encore en mode normal.

- 2nde étape, On met la carte eth0 en mode promiscuous en utilisant la commande « ifconfig eth0 promise ». Ensuite, on exécute les commandes ifconfig, ip link list ,ip address show et CPM. Ces commandes ont permis de détecter le mode promiscuous de la carte réseau eth0.

- 3ème étape, Maintenant, on désactive le mode promiscuous de la carte réseau eth0 en utilisant la commande « ifconfig eth0 -promise ».

- 4e étape, Maintenant que toutes les cartes réseau sont en mode normal, on exécute le sniffer Ethereal et on choisit comme interface réseau eth0. La différence avec la 1re étape, c'est qu'on va mettre l'interface réseau eth0 en mode promiscuous par un sniffer et non pas par une commande shell de configuration, telle que la commande « ifconfig eth0 promise ». Ensuite, on exécute les deux commandes ifconfig et CPM. Ces deux commandes n'ont pas pu détecter le mode promiscuous de l'interface réseau eth0. On passe ensuite aux commandes ip link list et ip address show. Le mode promiscuous de l'interface réseau eth0 a pu être détecté. Si le mode promiscuous a pu être détecté grâce aux commandes ip link list et ip address show et que les commandes ifconfig et CPM ont échoué, c'est que chaque commande possède un fonctionnement particulier. Par exemple, pour la commande ifconfig, il y a deux interprétations. La première invoque le flag système, appelé IFF Promise, qui est spécifique à l'interface réseau. Et c'est au niveau de ce flag que l'information sur le mode (promiscuous ou normal) de l'interface réseau existe. La commande ifconfig inspecte ce flag lors de son exécution. Donc, ifconfig ne reporte pas le mode promiscuous parce que le sniffer Ethereal ne touche pas au flag IFF_Promisc lors de son fonctionnement. En outre, Ethereal pourrait créer un flag interne jouant le même rôle que IFF Promise. Le flag IFF Promise se trouve donc intact et la commande ifconfig ne pourra pas s'apercevoir du mode promiscuous. Une autre approche incrimine la commande ifconfig et préconise par conséquent la méfiance quant aux résultats qu'elle affiche car elle ne reporte pas toujours l'état réel de l'interface réseau (<http://linux-ip.net/html/tools-ifconfig.html>). Nous sommes en mesure d'affirmer que les commandes les plus efficaces pour la détection locale du mode promiscuous sous Linux sont ip link list et ip address show.

Détection du mode promiscuous à l'aide des fichiers logs :

Sous Linux, on peut voir la date et l'heure exacte du début et de la fin de la mise en mode promiscuous d'une carte réseau. Pour cela, il suffit de lire le contenu du fichier « /var/log/messages ». Même si un administrateur n'arrive pas à détecter le mode promiscuous de la carte réseau locale en temps réel, la consultation du contenu de ce fichier lui permet de voir l'historique de la mise en mode promiscuous de la carte réseau et d'en tirer les conclusions adéquates.

4.4.4 – La méthode de latence

Dans un réseau, une machine qui écoute le trafic entrant provoque sur la machine un ralentissement. Si la machine distante est en mode promiscuous, les temps de réponse sont sensiblement plus longs. Une machine écoutant l'ensemble du trafic réseau, sera occupée, et mettra plus de temps à répondre.

La méthode de latence est un test qui fonctionne sur toutes les machines et tous les systèmes d'exploitation. Elle se déroule en deux temps. Dans une première phase, on mesure le temps de réponse moyen (round trip Unie – RTT) d'une machine cible. Dans une seconde phase, on inonde le réseau avec du trafic fictif et on mesure de nouveau le temps de réponse moyen de la machine cible. Le trafic fictif n'a aucun effet sur les machines en mode normal, mais a un effet conséquent sur les machines en mode promiscuous, particulièrement celles analysant les protocoles de la couche application pour capturer des logins et des mots de passe. Pour plus de précision, ce test doit être effectué plusieurs fois, en utilisant différentes méthodes de mesures.

Après la collecte des RTT avant et après l'envoi du trafic fictif, on utilise généralement un modèle statistique (tels que le z-Statistic et le Student-test) pour montrer que les deux échantillons des valeurs RTT collectés appartiennent à deux populations différentes et par conséquent que c'est le faux trafic qui en est la cause.

La méthode de latence est une méthode probabiliste et n'est donc pas sûre à 100 %, puisqu'elle peut générer des faux négatifs ou positifs. Cependant, combinée avec d'autres méthodes de détection, elle peut confirmer la présence d'un sniffer dans une machine. Comparée à d'autres méthodes de détection, elle est plus délicate à implémenter. D'un autre côté, elle peut dégrader de manière significative le bon déroulement du réseau et peut provoquer une congestion dans les machines exécutant des sniffers.

Pour certains systèmes d'exploitation, un éventuel problème est représenté par la possibilité que leurs noyaux traitent les paquets ping ICMP en priorité. Ainsi, les temps de réponse (RTT) des paquets ping resteront indépendants des charges des CPU causés par la réception de tout le trafic transitant sur le segment Ethernet, entraînant ainsi des faux positifs. Le choix du type de paquet ARP, ICMP, UDP ou TCP est donc décisif et dépend du système d'exploitation. Généralement, un paquet UDP ou TCP fonctionne sur tous les systèmes d'exploitation.

4.4.5 – La méthode physique

Un administrateur du réseau peut vérifier manuellement le hub ou le switch de son réseau pour voir s'il y a des raccordements inattendus. Par exemple, quelques commutateurs (switchers) ont un port spécial permettant la réception de tout le trafic circulant dans le réseau. Ce port a une multitude de noms, notamment «port mirroring », « monitoring port », « spanning port », « SPAN port » ou « link mode port ». Dans le cas des routeurs Cisco, ce port est appelé « SPANport ». Ainsi, si l'intrus arrive à connecter sa machine à ce port, il pourrait alors avec un sniffer espionner tout le trafic du réseau.

4.4.6 – La méthode du ping ICMP

Dans les noyaux des versions anciennes de quelques systèmes d'exploitation (Linux notamment), il y a une condition spécifique qui permet à un utilisateur de déterminer si une machine est en mode promiscuous ou non. Quand la carte réseau est placée en mode promiscuous, chaque paquet est passé directement au noyau du système d'exploitation. Quelques noyaux de systèmes d'exploitation vérifient seulement l'adresse destination IP dans le paquet pour déterminer si le paquet doit être traité ou non. Le système FreeBSD 4.11 en est un exemple.

Pour détecter le mode promiscuous dans les cartes réseau, les programmes de détection des sniffers (les anti-sniffers) exploitent cette spécificité. Un anti-sniffer transmet à une machine suspecte un paquet avec une fausse adresse destination MAC, mais avec une adresse destination IP valide. Le système avec une carte en mode promiscuous vérifie uniquement si l'adresse destination IP est valide. Un tel système va générer une réponse à ce type de paquet : la carte réseau est donc en mode promiscuous.

Le principe du ping ICMP est d'envoyer une fausse demande ping (ICMP Echo request) à une adresse IP valide d'une machine cible, mais avec une fausse adresse destination MAC. Si la machine est en mode normal (le filtre matériel activé), on s'attend à ce qu'elle ne réponde pas à cette demande. De même, si la machine est en mode promiscuous et que le filtre logiciel du noyau de son système d'exploitation filtre les paquets ICMP, elle ne va pas non plus répondre à cette demande. Les filtres matériel et logiciel de la machine n'acceptent en effet que les paquets dont l'adresse destination MAC est de type unicast, broadcast ou multicast. Par contre, si le mode est promiscuous et que le noyau du système d'exploitation ne filtre pas les paquets ICMP, la machine va générer un paquet de réponse ping, « ICMP echo reply ». Ceci peut être illustré par l'exemple suivant :

- La machine suspectée d'exécuter le sniffer a, par exemple, une adresse IP de X, et une adresse MAC de 00-40-05-A4-79-32.
- On change légèrement l'adresse MAC de la cible en 00-40-05-A4-79-33. Bien évidemment, on aura vérifié au préalable qu'aucune machine du réseau n'a cette adresse MAC.
- A partir d'une machine de test, on envoie la commande ping ICMP (ICMP echo request) avec l'adresse destination IP de X et la fausse adresse destination MAC de 00-40-05-A4-79-33.
- Puis, si l'on reçoit une réponse, on déduit que la machine suspecte n'exécute pas le filtrage matériel et le noyau de son système d'exploitation ne fait pas de filtrage logiciel, et par conséquent sa carte réseau est en mode promiscuous.
- On répète ce processus pour toutes les machines du réseau..

Il est important de noter que si une machine ne génère pas de réponse, ceci n'implique pas systématiquement que sa carte réseau est en mode normal. Une machine ne répond pas parce que le noyau de son système d'exploitation

exécute du filtrage logiciel, en dépit du mode promiscuous. Notons que les noyaux des nouvelles versions de la plupart des systèmes d'exploitation exécutent des filtres logiciels. Au lieu d'utiliser les paquets ping ICMP, on peut utiliser :

- tout protocole qui peut générer des paquets de réponse, par exemple, le protocole TCP (une demande de connexion TCP), ou le protocole UDP (un echo request sur le port 7) ;
- tout protocole qui peut générer des paquets d'erreur destinés à la machine source, par exemple un paquet comportant un entête IP avec de fausses valeurs.

De nos jours, cette technique est largement connue, et les agresseurs parviennent à développer du filtrage virtuel dans les codes de leurs sniffers. Ainsi, en plus de sa fonction principale de capture des paquets circulant dans un réseau, le sniffer bloquera les paquets ping ICMP portant des adresses destination MAC qui ne sont pas de type unicast, broadcast ou multicast. De cette manière, la machine exécutant le sniffer ne peut pas générer de réponses à ces paquets. De plus, les noyaux des nouvelles versions de la plupart des systèmes d'exploitation incluent des filtres logiciels. Par conséquent, la méthode du ping ICMP et même toutes les autres méthodes utilisant d'autres protocoles, ne permettront plus la détection des machines dont les cartes réseau sont en mode promiscuous.

4.4.7 – La méthode de l'ARP

La méthode ARP est semblable à la méthode du ping ICMP, mais un faux paquet ARP (ARP request) est employé à la place d'un faux paquet ping ICMP. Le principe général de cette méthode est de tester le réseau machine par machine en envoyant une requête ARP avec une fausse adresse destination MAC mais avec une adresse destination IP valide. Comme nous nous intéressons aux réseaux Ethernet partagés, ce paquet devrait passer devant toutes les machines liées au même segment Ethernet.

- Si la machine est en mode normal (le filtre matériel activé), nous attendons à ce qu'elle ne va pas répondre à cette demande. Egalement, si la machine est en mode promiscuous et le filtre logiciel du noyau de son système d'exploitation filtre les paquets ARP, nous attendons à ce qu'elle ne va pas répondre à cette demande. Ceci parce que les filtres matériel et logiciel de la machine acceptent seulement les paquets dont l'adresse destination MAC est de type unicast, broadcast ou multicast. Par contre, si la machine est en mode promiscuous et le filtre logiciel du noyau de son système d'exploitation ne filtre pas les paquets ARP, nous attendons à ce qu'elle va répondre avec un paquet de type réponse ARP (ARP reply) pour fournir son adresse MAC supposée recherchée.
- A l'instar de la méthode du ping ICMP, actuellement la plupart des systèmes d'exploitation intègrent au niveau de leurs noyaux des filtres logiciels pour filtrer les paquets ARP de tel sorte la méthode de l'ARP ne peut plus détecter les cartes réseau en mode promiscuous. Egalement,

cette technique est maintenant largement connue. Les intrus peuvent développer du filtrage virtuel dans les codes de leurs sniffers. Ainsi, en plus de sa fonction principale de capture des paquets circulant dans un réseau, le sniffer filtre les paquets ARP portant des adresses MAC de destination qui ne sont pas de type unicast, broadcast et multicast. De la sorte, la machine exécutant le sniffer ne peut pas générer des réponses ARP à des fausses requêtes ARP.

4.4.8 – La méthode de l'attaque du cache ARP

Principe de l'attaque de corruption du cache ARP :

Ce paragraphe présente l'attaque de corruption du cache ARP (ARP cache poisoning), qui exploite les vulnérabilités du protocole ARP. Cette attaque est applicable uniquement sur un réseau Ethernet exécutant IP.

L'ARP fonctionne en envoyant des demandes ARP (ARP request). Une demande ARP pose la question « Votre adresse IP est-elle x.x.x.x ? Si oui, envoyez votre adresse MAC ». Ces demandes sont émises à tous les ordinateurs sur le réseau LAN (paquets de diffusion), même s'il s'agit d'un réseau commuté. Chaque ordinateur examine la demande ARP, vérifie s'il est, à ce moment-là, assigné à l'adresse IP indiquée: Si oui, il envoie une réponse ARP (ARP reply) contenant son adresse MAC.

Pour réduire au minimum le nombre de demandes (ou réponses) ARP émises, les systèmes d'exploitation gardent un cache des réponses ARP (le cache ARP). La mise à jour du cache ARP peut se faire de deux manières. Dans le premier cas de figure, quand un ordinateur reçoit une réponse ARP, il va mettre à jour son cache ARP avec la nouvelle association d'IP/MAC correspondant à l'émetteur de la réponse ARP. Des systèmes d'exploitation tels que Windows 2000 et FreeBSD 4.11 mettront à jour leurs caches ARP à la réception de réponses ARP, même si aucune requête ARP n'a pas été envoyée. Par contre, des systèmes tels que Windows XP, Linux 2.4 et 2.6 ne mettront pas à jour leurs caches ARP suite à la réception de réponses ARP, s'ils n'ont pas déjà envoyé des requêtes ARP. Dans le second cas, quand un ordinateur reçoit une requête ARP, il va mettre à jour son cache ARP avec l'association d'IP/MAC correspondant à l'émetteur de la requête ARP. Tous les systèmes d'exploitation testés mettront à jour leurs caches ARP à la réception d'une requête ARP même si l'entrée n'existe pas dans le cache. Le tableau 5.6 montre le résultat des tests de mise à jour des caches ARP de quelques systèmes, sous les conditions suivantes :

- une entrée existe dans le cache ARP et le système reçoit une requête ARP visant à la mettre à jour ;
- une entrée existe dans le cache ARP et le système reçoit une réponse ARP visant à la mettre à jour ;
- l'entrée n'existe pas dans le cache ARP et le système reçoit une requête ARP visant à la créer ;
- l'entrée n'existe pas dans le cache ARP et le système reçoit une réponse

ARP visant à la créer.

Les résultats des tests amènent les conclusions suivantes :

- si l'entrée n'existe pas dans le cache ARP, tous les OS à l'exception de Windows 2000 et de FreeBSD 4.11, n'autorisent pas la création de l'entrée par une réponse ARP ;
- si l'entrée n'existe pas dans le cache ARP, tous les systèmes autorisent la création d'une entrée par une requête ARP ;
- par contre, si l'entrée existe déjà dans le cache ARP, tous les systèmes autorisent sa mise à jour par une réponse ARP (même en absence auparavant d'une requête ARP) ou par une requête ARP.

L'attaque de corruption du cache ARP consiste à mettre à jour les caches ARP des machines cibles avec de fausses entrées IP/MAC, en utilisant des paquets ARP. Si une entrée n'existe pas dans le cache ARP, pour la plupart des systèmes, une réponse ARP ne permet pas de la créer. Par contre, pour tous les autres systèmes, une requête ARP permet de créer une entrée inexistante dans le cache ARP. Par conséquent, pour mettre à jour les caches ARP des machines cibles avec de fausses entrées IP/MAC, seules des requêtes ARP falsifiées peuvent être utilisées. En envoyant de fausses requêtes ARP, contenant des fausses adresses sources IP (IP_X) et MAC (MAC_X), un ordinateur cible B mettra à jour son cache ARP avec ces adresses. Le processus de mise à jour du cache ARP d'une machine cible avec une entrée IP/MAC falsifiée est désigné sous le nom de « corruption ARP » (ARP poisoning). L'outil Winarp_sk, téléchargeable à partir de l'adresse www.arp-sk.org, ou un générateur de paquet tel que celui du sniffer CommView, ou FRAMEIP (lien vers EXE) permettent d'envoyer des paquets de requêtes ARP falsifiées.

Détection des cartes en mode promiscuous :

La méthode proposée pour la détection des cartes réseau en mode promiscuous se base essentiellement sur l'attaque de corruption du cache ARP et utilise trois phases :

- dans la première phase, nous utilisons l'attaque de corruption du cache ARP pour corrompre seulement les caches ARP des machines du réseau dont les cartes sont en mode promiscuous. L'attaque utilise une fausse entrée IP/MAC appelée IP-Test/MAC-Test ;
- dans la deuxième phase, nous essayons d'établir une connexion TCP avec chacune des machines du réseau sur un port quelconque, ouvert ou fermé ;
- dans la dernière phase, nous utilisons un sniffer afin de capturer tout paquet contenant la fausse entrée IP-Test/MAC-Test.

Nous démontrerons que les machines qui généreront des paquets TCP contenant cette fausse entrée IP-Test/MAC-Test sont en mode promiscuous. Cependant, les machines qui génèrent des requêtes ARP dans le but de chercher l'adresse MAC de la machine d'adresse IP IP-Test ne sont pas en

mode promiscuous. Nous allons décrire en détail les trois phases précédentes en utilisant une machine nommée machine de test.

Phase 1 : l'attaque de corruption du cache ARP. Le but de cette phase est de corrompre seulement les caches ARP des machines dont les cartes sont en mode promiscuous. Pour ce faire, nous envoyons pour chaque machine du réseau une fausse requête ARP piège avec dans les entêtes Ethernet et ARP les champs suivants :

Entête Ethernet

Adresse source MAC = N'importe quelle adresse

Adresse destination MAC = FF:FF:FF:FF:FF:FF (B47)

Type Ethernet = 0x806 (Protocole ARP)

Entête ARP

Type ARP = 0x01 (requête ARP)

Adresse source MAC = Fausse adresse MAC (MAC-Test)

Adresse source IP = Fausse adresse IP (IP-Test)

Adresse destination MAC = 00:00:00:00:00:00

Adresse destination IP = Adresse IP de la machine cible

- Si une machine dans le réseau cible est en mode normal, la fausse requête ARP piège sera bloquée par le filtre matériel, puisque l'adresse destination MAC dans l'entête Ethernet est la fausse adresse de broadcast. Par conséquent, la mise à jour du cache ARP de cette machine ne peut être réalisée.

- Par contre, si la machine est en mode promiscuous, le filtre matériel est désactivé, mais le filtre logiciel est encore activé. Par conséquent, la fausse requête ARP piège est envoyée directement au filtre logiciel qui va l'accepter puisqu'elle a la fausse adresse MAC de broadcast comme adresse destination MAC. Ainsi, le cache ARP de cette machine sera mis à jour par une nouvelle fausse entrée, à savoir : IP-Test/MAC-Test. Notons que si nous choisissons une adresse MAC de broadcast générale (FF:FF:FF:FF:FF:FF) comme adresse destination MAC au niveau de l'entête Ethernet de la fausse requête ARP piège, alors tous les caches ARP des machines du réseau cible seraient corrompus par l'attaque de corruption du cache ARP. Ainsi, une telle adresse devrait être écartée parce qu'elle ne permet pas de détecter le mode promiscuous des cartes réseau.

En conclusion, cette première phase nous a permis de corrompre uniquement les caches ARP des machines dont leurs cartes sont en mode promiscuous.

Phase 2 : demande d'établissement d'une connexion TCP piège. Une fois la fausse entrée IP-Test/MAC-Test créée dans les caches ARP des machines en mode promiscuous, nous essayerons d'établir une connexion TCP piège avec chaque machine du réseau cible, sur un port quelconque. Il est important de noter que le port choisi peut être un port ouvert ou fermé. En général, pour établir une connexion TCP avec une machine cible, il est nécessaire d'envoyer à la machine un paquet TCP SYN avec le SYN=1. En outre, la valeur du champ de l'adresse source IP dans l'entête IP du paquet TCP SYN est égale à l'adresse

IP de la machine qui demande l'établissement de la connexion. Nous allons toutefois envoyer un paquet TCP SYN piège contenant une fausse adresse source IP dans l'entête IP à chaque machine du réseau pour demander l'établissement d'une connexion TCP.

Ce paquet TCP SYN piège présente la particularité suivante : l'adresse source IP dans l'entête IP du paquet TCP SYN est égale à la fausse adresse IP-Test. En principe, elle doit être égale à l'adresse IP de la machine de test, puisque c'est cette dernière qui demande l'établissement de la connexion TCP. Les valeurs des champs du paquet TCP SYN piège utilisé pour établir la connexion TCP avec chaque machine cible dans le réseau sont :

Entête Ethernet

Adresse source MAC = Adresse physique de la machine de test

Adresse destination MAC = Adresse physique de la machine cible

Type Ethernet = 0x800 (le protocole IP)

Entête IP

Adresse source IP = La fausse adresse IP (IP-Test)

Adresse destination IP = L'adresse IP de la machine cible

Entête TCP

Port source = N'importe quelle valeur entre 1 et 65 535

Port destination = N'importe quelle valeur entre 1 et 65 535

Flag = 0x02 (SYN)

Phase 3 : Détection des machines avec des cartes réseau en mode promiscuous – Analyse des résultats. Une fois les fausses requêtes ARP pièges (phase 1) et les paquets TCP SYN pièges (phase 2) envoyés aux différentes machines du réseau, nous attendons les réponses possibles suivantes :

- une requête ARP : si une machine est en mode normal, lorsqu'elle reçoit le paquet TCP Syn piège dont le champ adresse destination IP au niveau de l'entête IP correspond à sa propre adresse IP, elle va croire que la machine dont l'adresse IP IP-Test veut converser avec elle. Comme aucune entrée correspondante à l'adresse IP-Test n'existe dans son cache ARP (phase 1), la machine cible enverra une requête ARP qui s'interroge sur l'adresse MAC qui correspond à IP-Test. Cette machine ne peut donc avoir une carte en mode promiscuous ;
- un paquet TCP Syn/Ack : si une machine cible est en mode promiscuous et si le port de destination dans l'entête TCP du paquet TCP Syn piège est égal à un port ouvert dans la machine, lorsqu'elle reçoit le paquet TCP Syn piège, elle générera un paquet TCP Syn/Ack dont l'adresse destination MAC est égal à MAC-Test et l'adresse destination IP est égal à IP-Test. Or, une question se pose : comment cette machine a-t-elle pu avoir l'adresse MAC MAC-Test ? La seule explication est que son cache ARP contient la fausse entrée IP-Test/MAC-Test. Donc, son cache ARP a été corrompu lors de l'attaque de corruption du cache ARP durant la Phase 1. Par conséquent, cette machine a une carte réseau en mode promiscuous ;
- un paquet TCP Reset : si une machine cible est en mode promiscuous et

si le port de destination dans l'entête TCP du paquet TCP Syn piège est égal à un port fermé dans la machine, lorsque elle reçoit le paquet TCP Syn piège, elle générera un paquet TCP Rst (indiquant que la connexion TCP ne peut pas être établie car le port de destination n'est pas accessible). Ce paquet TCP Rst a un entête Ethernet dont l'adresse destination MAC est égal à MAC-Test et a un entête IP dont l'adresse destination IP est égale à IP-Test. Or, une question simple, comment cette machine a pu avoir l'adresse MAC MAC-Test ? La seule explication est que son cache ARP contient la fausse entrée IP-Test/MAC-Test. Donc son cache ARP a été corrompu lors de l'attaque de l'empoisonnement du cache ARP durant la phase 1. Par conséquent, cette machine a une carte réseau en mode promiscuous.

En conclusion, une machine qui génère un paquet de réponse TCP avec les fausses adresses MAC-Test et IP-Test respectivement comme adresse destination MAC et adresse destination IP, a sûrement la carte en mode promiscuous. Son cache ARP contient la fausse entrée IP-Test/MAC-Test. Cette fausse entrée a été créée suite à l'attaque de corruption du cache ARP durant la phase 1. Cependant, une machine dont le cache ARP n'est pas corrompu générera une requête ARP dans le but de rechercher l'adresse MAC qui correspond à la fausse adresse IP IP-Test. Si un tel paquet est reçu, la carte réseau de la machine émettrice n'est pas en mode promiscuous. Il est important de noter que si la machine cible autorise la génération des réponses ARP à la suite de la réception de requêtes ARP.

4.5 – Les outils de détection des sniffers

Les outils de détection des sniffers, appelés anti-sniffers, utilisent pratiquement la plupart des techniques de détection décrites avant. Les anti-sniffers les plus connus sont listés dans le tableau ci-dessous :

4.5.1 – LOpht AntiSniff

C'est un ancien outil qui est capable de détecter les sniffers exécutés sous d'anciennes versions des systèmes d'exploitation (tels que Windows 98 et Windows 95) en utilisant principalement les méthodes DNS, ARP et ping ICMP.

LOpht AntiSniff utilise deux modes de détection, à savoir les techniques liées aux systèmes d'exploitation et les techniques du calcul de latence.

AntiSniff a été conçu pour être exécuté de deux façons. Premièrement, pour une « analyse du réseau » pour identifier rapidement quelles machines sur le réseau local sont les plus susceptibles d'être étudiées dans la seconde étape. En second lieu, AntiSniff peut être exécuté de base pour la détection de sniffer, balayant le réseau à intervalles programmées et régulières, comparant les tests de réponses des machines sur une bases de temps et en positionnant des alarmes basées sur des événements définis pour l'utilisateur et en vérifiant ces réponses (test sur arp, dns, echo icmp...)

4.5.2 – The sentinel

The Sentinel fonctionne sur les noyaux *BSD et Linux. Les méthodes de détection sont l'ARP, le DNS et l'écho ICMP

4.5.3 – Anasil Network Analyser

Anasil est un utilitaire d'analyse de réseau Ethernet. Il est destiné au contrôle des liens réseaux, à la création et à la maintenance d'une liste d'ordinateurs actifs en réseau, aux tests du lien réseau et de la connexion entre les postes. Il possède en plus des méthodes de détection de sniffers.

4.5.4 – PromiScan

PROMISCan est un logiciel qui permet de détecter à distance, au sein d'un réseau local, quelles sont les machines dont l'interface réseau est en mode promiscuous.

5 – Leurrer les anti-sniffers

Les anti-sniffers fonctionnent comme la plupart des outils de détection, et ils sont loin d'être parfaits. Il existe plusieurs méthodes pour leurrer les anti-sniffers en rendant les sniffers indétectables. Il est important de savoir paramétrer correctement un sniffer pour écouter le trafic entrant et être totalement passif (éviter les requêtes DNS inverses par exemple). En outre, le filtrage du trafic entrant et/ou sortant permet de rendre inopérantes plusieurs techniques de détection. Dans ce qui suit, nous citerons quelques démarches à réaliser pour rendre les sniffers indétectables par les anti-sniffers :

- ne pas générer de requêtes DNS. Pour ce faire, il faut désactiver l'option « DNS resolving » dans les options du sniffer.
- ne pas générer de réponses ARP. Dans ce cas, il faudrait choisir l'une des méthodes suivantes :
 - modifier le noyau du système d'exploitation pour ne plus générer des réponses ARP. Ceci n'est possible qu'avec les systèmes d'exploitation libres (open source), par exemple Linux. Le fichier « /usr/src/linux/ /net/ipv4/arp.c » dans Linux contient le code du protocole ARP,
 - désactiver le protocole ARP. Par exemple sous Linux, exécuter la commande « ifconfig eth0 -arp » permet de désactiver complètement le protocole ARP de la pile TCP/IP et par conséquent empêcher la carte réseau de répondre aux requêtes ARP.

Le cache ARP de la machine n'a aucune entrée et donc la machine ne peut plus dialoguer avec le réseau mais elle peut toujours exécuter un sniffer. Par contre, ceci peut mettre cette machine en position suspecte puisqu'elle est présente sur le réseau alors qu'elle ne répond pas aux requêtes ARP. Sous Windows, il n'existe pas de commande en ligne permettant de désactiver le protocole ARP.

- utiliser un firewall pour filtrer les paquets ARP entrants et sortants, par exemple le. firewall Kerio Personal Firewall.
- utiliser un noyau mis à jour ou modifié pour corriger le problème mentionné pour la plupart des systèmes d'exploitation concernant le filtrage logiciel des fausses adresses de broadcast et de multicast. Ainsi le filtre logiciel du noyau du système bloquera tout paquet avec de telles fausses adresses ;
- bloquer tout le trafic entrant et sortant de telle sorte la machine exécutant le sniffer soit déconnectée complètement du réseau. Donc, pratiquement toutes les méthodes de détection, notamment la méthode du ARP, la méthode du DNS, la méthode de latence, la méthode de l'attaque du cache ARP, ne permettent plus la détection des sniffers. Ceci peut se faire simplement grâce à un firewall installé sur la machine qui exécute le sniffer.
- Cesser le sniffer quand le trafic réseau excède un certain taux. Ceci peut être une indication qu'un anti-sniffer est en train d'injecter du trafic dans le réseau lorsqu'il utilise la méthode de latence.

Un sniffer appelé Anti AntiSniffer a été réalisé juste après la sortie d'Antisniff. Ce nouveau sniffer utilise quelques-unes des techniques décrites pour éviter d'être découvert par AntiSniff ou un outil similaire. Un autre Anti-AntiSniffer « Enigma », de www.netninja.com, propose aussi de rendre l'écoute d'un réseau sûre et indétectable. Dans l'attente d'un « anti-anti-anti-sniffer », la bataille n'est pas finie !