La sécurité dans l'environnement IP

Les échanges de données s'effectuent généralement entre un client et un serveur, mais les applications peer-to-peer, qui vont directement d'un client à un autre client, deviennent de plus en plus courantes. En règle générale, le client se sert d'un identifiant déterminé par un login et un mot de passe mis en clair sur le réseau. Le client est donc identifié, et il obtient l'autorisation d'accéder au serveur grâce à son mot de passe. Cependant, cette solution peut se révéler bien faible face à des pirates. L'absence d'authentification de la provenance des paquets IP rend possible de nombreuses attaques, comme les dénis de service *(denial of service)*, c'est-à-dire le refus ou l'impossibilité pour un serveur de fournir l'information demandée.

Il existe de nombreuses familles d'attaques dans le réseau Internet. Ce chapitre commence par donner un certain nombre d'exemples de ces attaques avant d'examiner les parades possibles.

Les attaques par Internet

Les attaques concernent deux grands champs : celles qui visent les équipements terminaux et celles qui visent le réseau Internet lui-même. Elles ne sont pas totalement décorrélées puisque les attaques des machines terminales par Internet utilisent souvent des défauts d'Internet.

Les attaques du réseau Internet lui-même consistent à essayer de dérégler un équipement de routage ou un serveur, comme les serveurs DNS, ou à obstruer les lignes de communication. Les attaques des machines terminales consistent à prendre le contrôle de la machine pour effectuer des opérations non conformes. Très souvent, ces attaques s'effectuent par le biais des logiciels réseau qui se trouvent dans la machine terminale. Cette section explicite quelques attaques parmi les plus classiques.

Les attaques par ICMP

Le protocole ICMP (Internet Control Message Protocol) est utilisé par les routeurs pour transmettre des messages de supervision permettant, par exemple, d'indiquer à un utilisateur la raison d'un problème. Un premier type d'attaque contre un routeur ou un serveur réseau consiste à générer des messages ICMP en grande quantité et à les envoyer vers la machine à attaquer à partir d'un nombre de sites important.

Pour inonder un équipement de réseau, le moyen le plus simple est de lui envoyer des messages de type ping lui demandant de renvoyer une réponse. On peut également inonder un serveur par des messages de contrôle ICMP d'autres types.

Les attaques par TCP

Le protocole TCP travaille avec des numéros de port qui permettent de déterminer une adresse de socket, c'est-à-dire d'un point d'accès au réseau. Cette adresse de socket est formée par la concaténation de l'adresse IP et de l'adresse de port. À chaque application correspond un numéro de port, par exemple 80 pour une application HTTP.

Une attaque par TCP revient à utiliser un point d'accès pour faire autre chose que ce pour quoi il a été défini. En particulier, un pirate peut utiliser un port classique pour entrer dans un ordinateur ou dans le réseau d'une entreprise. La figure 34.1 illustre une telle attaque. L'utilisateur ouvre une connexion TCP sur un port correspondant à l'application qu'il projette de dérouler. Le pirate commence à utiliser le même port en se faisant passer pour l'utilisateur et se fait envoyer les réponses. Éventuellement, le pirate peut prolonger les réponses vers l'utilisateur de telle sorte que celui-ci reçoive bien l'information demandée et ne puisse se douter de quelque chose.

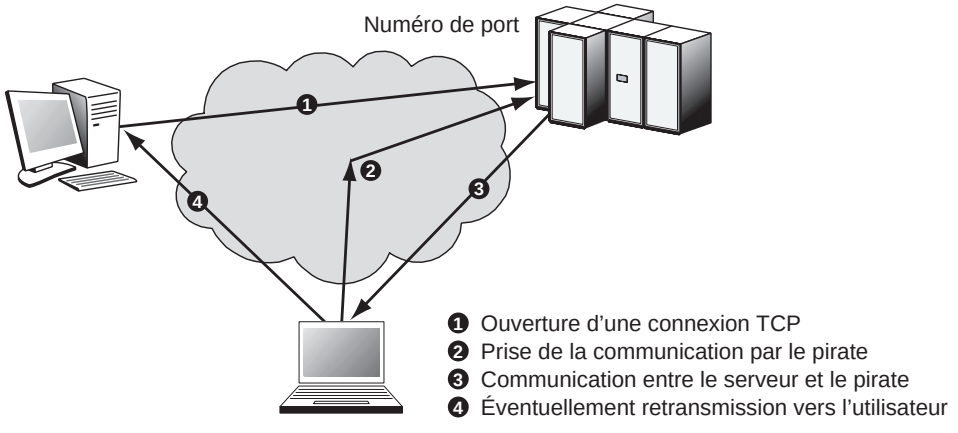


Figure 34.1
Attaque par le protocole TCP

Nous verrons en fin de chapitre comment les pare-feu, ou firewalls, essaient de parer ce genre d'attaque en bloquant certains ports.

Les attaques par cheval de Troie

Dans l'attaque par cheval de Troie, le pirate introduit dans la station terminale un programme qui permet de mémoriser le login et le mot de passe de l'utilisateur. Ces informations sont envoyées à l'extérieur par le biais d'un message vers une boîte aux lettres anonyme.

Diverses techniques peuvent être utilisées pour cela, allant d'un programme qui remplace le gestionnaire de login jusqu'à un programme pirate qui espionne ce qui se passe dans le terminal.

Les attaques par dictionnaire

Beaucoup de mots de passe étant choisis dans le dictionnaire, il est très simple pour un automate de les essayer tous. De nombreuses expériences ont démontré la facilité de cette attaque et ont mesuré que la découverte de la moitié des mots de passe des employés d'une grande entreprise s'effectuait en moins de deux heures.

Une solution simple pour remédier à cette attaque est de complexifier les mots de passe en leur ajoutant des lettres majuscules, des chiffres et des signes comme !, ?, &, etc.

Les autres attaques

Le nombre d'attaques possibles est bien trop grand pour que nous puissions les citer toutes. De plus, de nouvelles procédures d'attaque s'inventent chaque jour.

Les attaques par écoute consistent, pour un pirate, à écouter une ligne de communication et à interpréter les éléments binaires qu'il intercepte. Les attaques par fragmentation utilisent le fait que les informations de supervision se trouvent dans la première partie du paquet à un emplacement parfaitement déterminé. Un pirate peut modifier la valeur du bit de fragmentation, ce qui a pour effet de faire croire que le message se continue alors qu'il aurait dû se terminer. Le pare-feu voit donc arriver une succession de fragments qui suivent les fragments de l'utilisateur sans se douter que ces fragments complémentaires ont été ajoutés par le pirate.

Les algorithmes de routage sont à la base de nombreuses attaques. En effectuant des modifications sur les tables de routage, le pirate peut récupérer de nombreuses informations qui ne lui sont pas destinées ou dérouter les paquets, lesquels, par exemple, vont effectuer des boucles et saturer le réseau.

De la même façon, de nombreuses attaques sont possibles en perturbant un protocole comme ARP (Address Resolution Protocol), soit pour prendre la place d'un utilisateur, soit en captant des données destinées à un autre.

Les parades

Les parades aux attaques sont nombreuses. Elles relèvent autant du comportement humain que de techniques spécifiques. Nous allons examiner les principales : l'authentification, l'intégrité du flux, la non-répudiation, la confidentialité du flux et la confidentialité au niveau de l'application.

L'authentification

Une première parade visant à empêcher qu'un autre terminal que celui prévu ne se connecte ou bien qu'un terminal ne se connecte sur un serveur pirate est offerte par les méthodes d'authentification. L'authentification peut être simple, et ne concerner que l'utilisateur, ou mutuelle, et impliquer à la fois le client et le serveur.

Dans des applications de type Telnet, e-mail ou LDAP, le client s'authentifie avec un mot de passe auprès du serveur pour établir ses droits. Dans une application de commerce électronique (HTTP), il est nécessaire d'authentifier le serveur puis le client, généralement à l'aide d'un mot de passe. Le protocole HTTP ne possédant pas de moyen efficace d'authentification du client, la société Netscape a introduit vers 1995 la notion de cookie, destinée à identifier un flux de requêtes HTTP disjointes.

L'intégrité du flux de données

L'intégrité d'un flux de données demande qu'il ne puisse y avoir une altération des informations transportées. Un pirate pourrait en effet modifier une information pour tromper le récepteur. Il est à noter qu'intégrité ne signifie pas confidentialité. En effet, il est possible que l'information ne soit pas confidentielle et qu'elle puisse être recopiée, sans que cela pose de problème à l'utilisateur. Cependant, l'utilisateur veut que son information arrive intègre au récepteur.

La solution classique à ce genre de problème consiste à utiliser une empreinte. À partir de l'ensemble des éléments binaires dont on souhaite assurer l'intégrité, on calcule une valeur, qui ne peut être modifiée sans que le récepteur s'en rende compte. Les empreintes regroupent les solutions de type empreinte digitale, signature électronique, analyse rétinienne, reconnaissance faciale et, d'une manière générale, tout ce qui permet de signer de façon unique un document. Ces différentes techniques de signature proviennent de techniques d'authentification puisque, sous une signature, se cache une authentification.

Dans les réseaux IP, la pratique de la signature électronique est de plus en plus mise en œuvre pour faciliter le commerce et les transactions financières.

La non-répudiation

La non-répudiation consiste à empêcher l'éventuel refus d'un récepteur d'effectuer une tâche suite à un démenti de réception. Si la valeur juridique d'un fax est reconnue, celle d'un message électronique ne l'est pas encore. Pour qu'elle le soit, il faut un système de non-répudiation. Les parades visant à éviter qu'un utilisateur répudie un message reçu proviennent essentiellement d'une signature unique sur le message et sur son accusé de réception, c'est-à-dire une signature qui ne serait valable qu'une seule fois et serait liée à la transmission du message qui a été répudié. Un système de chiffrement à clés publiques peut être utilisé dans ce contexte.

Une autre solution, qui se développe, consiste à passer par un notaire électronique, qui, par un degré de confiance qui lui est attribué, peut certifier que le message a bien été envoyé et reçu.

Une difficulté importante de la non-répudiation dans une messagerie électronique provient de la vérification que le récepteur en a pris possession et a lu le message. Il n'existe pas de règle aujourd'hui sur Internet pour envoyer des messages de type lettre recommandée. Le récepteur peut, par exemple, recevoir le message dans sa boîte aux lettres électronique mais ne pas le récupérer. Il peut également recopier le message dans la boite aux lettres de son terminal et le supprimer sans le lire.

Les techniques de non-répudiation ne sont pas encore vraiment développées dans le monde IP. En effet, cette fonction de sécurité est souvent jugée moins utile que les autres. Cependant, elle est loin d'être absente. En effet, dans le commerce électronique elle est capitale pour qu'un achat ne puisse être décommandé sans certaines conditions déterminées dans le contrat d'achat. Cette fonction serait également utile dans des applications telles que la messagerie électronique, où l'on aimerait être sûr qu'un message est bien arrivé.

Même si la non-répudiation n'est pas implémentée de façon automatique, elle est proposée dans de nombreuses applications qui en ont besoin.

La confidentialité

La confidentialité désigne la capacité de garder une information secrète. Le flux, même s'il est intercepté, ne doit pas pouvoir être interprété. La principale solution permettant d'assurer la confidentialité d'un flux consiste à le chiffrer. Les systèmes de chiffrement ont été présentés au chapitre 33.

Aujourd'hui, étant donné la puissance des machines qui peuvent être mises en jeu pour casser un code, il faut utiliser de très longues clés. Les clés de 40 bits peuvent être percées en quelques secondes et celles de 128 bits en quelques minutes sur une très grosse machine. Une clé RSA de 128 bits a été cassée en quelques heures par un ensemble de machines certes important mais accessible à une entreprise.

Pour casser une clé, il faut récupérer des données chiffrées, parfois en quantité importante, ce qui peut nécessiter plusieurs heures d'écoute, voire plusieurs jours si la ligne est à faible débit. Une solution à ce problème de plus en plus souvent utilisée consiste à changer de clé régulièrement de telle sorte que l'attaquant n'ait jamais assez de données disponibles pour casser la clé.

Dans la réalité, il est plus facile de pirater une clé que d'effectuer son déchiffrement. Une parade pour contrer les pirates réside dans ce cas dans un contrôle d'accès sophistiqué des bases de données de clés.

La confidentialité est aujourd'hui un service fortement utilisé dans le monde IP. IPsec en est un très bon exemple, et nous le détaillons un peu plus loin dans ce chapitre. De nouvelles méthodes, comme le chiffrement quantique, sont à l'étude et pourraient déboucher sur des méthodes encore plus sûres.

La sécurité dans les protocoles

Conçus avant les années 2000, les protocoles du monde IP n'ont pas intégré de fonctions de sécurité. De nombreuses failles de sécurité existent donc, qui sont comblées régulièrement par des RFC spécifiques.

Les attaques sur les protocoles de gestion ou de contrôle peuvent facilement arrêter le fonctionnement d'un réseau. Il suffit, par exemple, de faire croire aux accès que le réseau est saturé ou que les nœuds sont en panne pour que les performances du réseau s'effondrent totalement.

La sécurité dans SNMP

La RFC 2274 définit le modèle USM (User-based Security Model) de sécurité de SNMP, qui offre à la fois une authentification et un service de sécurité.

Les principales attaques dont SNMP peut être l'objet sont les suivantes :

- Modification de l'information : une entité peut altérer un message en transit généré par une entité autorisée pour modifier une opération de type comptabilité, configuration ou opération.
- Mascarade : une entité prend l'identité d'une entité autorisée.
- Modification à l'intérieur d'un flot de messages : SNMP est construit pour gérer un protocole de transport en mode sans connexion. Les messages peuvent être réordonnés d'une façon différente de celle d'origine et détruits ou rejoués d'une autre manière. Par exemple, un message qui redémarre une machine peut être copié puis rejoué ultérieurement.
- Ordre de secret : une entité peut observer les échanges entre un manager et son agent et apprendre les valeurs des objets gérés. Par exemple, l'observation d'un ensemble de commandes capables de modifier un mot de passe permettrait à un utilisateur de modifier le mot de passe et d'attaquer le site.

Le modèle de sécurité USM ne prend pas en compte les deux fonctionnalités suivantes :

- Refus de service : un attaquant interdit l'échange d'informations entre un manager et son agent. Nous avons vu au chapitre 29, consacré à la gestion de réseau, que les échanges d'information de gestion s'effectuaient entre un manager de gestion et ses agents. Si le manager ne reçoit plus les informations du réseau et *vice versa*, les agents ne reçoivent plus les commandes du manager, et le processus de gestion du réseau ne peut plus s'effectuer. On appelle cette attaque un refus de service, puisque le service de gestion refuse de travailler.
- Analyse de trafic : un attaquant observe le type de trafic qui s'effectue entre un manager et son agent. L'analyse permet de détecter les ordres qui sont passés et les remontées d'information. Après analyse du trafic, le pirate peut faire croire au manager que le trafic est totalement différent de ce qu'il est effectivement dans le réseau.

Pour contrer ces différentes attaques, deux fonctions cryptographiques ont été définies dans USM : l'authentification et le chiffrement. Pour les réaliser, le moteur SNMP requiert deux valeurs : une clé privée et une clé d'authentification. Ces valeurs sont des attributs de l'utilisateur qui ne sont pas accessibles par des primitives SNMP.

Deux algorithmes d'authentification sont disponibles : HMAC-MD5-96 et HMAC-SHA-96. L'algorithme HMAC utilise une fonction de hachage sécurisée et une clé secrète pour produire un code d'authentification du message. Ce protocole fortement utilisé dans Internet est décrit en détail dans la RFC 2104.

IPsec (IP sécurisé)

Le monde TCP/IP permet d'interconnecter plusieurs millions d'utilisateurs, lesquels peuvent souhaiter que leur communication reste secrète. Internet transporte de plus un grand nombre de transactions de commerce électronique, pour lesquelles une certaine confidentialité est nécessaire, par exemple pour prendre en charge la transmission de numéros de carte bancaire.

L'idée développée dans les groupes de travail sur la sécurité du commerce électronique dans le monde IP consiste à définir un environnement contenant un ensemble de mécanismes de sécurité. Les mécanismes de sécurité appropriés sont choisis par une association de sécurité *(voir ci-après)*. En effet, toutes les communications n'ont pas les mêmes caractéristiques, et leur sécurité ne demande pas les mêmes algorithmes.

Chaque communication se définit par sa propre association de sécurité. Les principaux éléments d'une association de sécurité sont les suivants :

- algorithme d'authentification ou de chiffrement utilisé ;
- clés globales ou spécifiques à prendre en compte ;
- autres paramètres de l'algorithme, comme les données de synchronisation ou les valeurs d'initialisation ;
- durée de validité des clés ou des associations ;
- sensibilité de la protection apportée (secret, top secret, etc.).

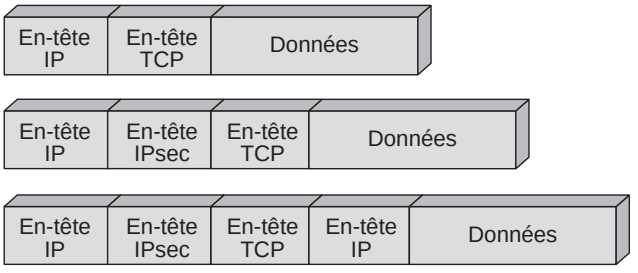
La solution IPsec introduit des mécanismes de sécurité au niveau du protocole IP, de telle sorte qu'il y ait indépendance vis-à-vis du protocole de transport. Le rôle de ce protocole de sécurité est de garantir l'intégrité, l'authentification, la confidentialité et la protection contre les techniques rejouant des séquences précédentes. L'utilisation des propriétés d'IPsec est optionnelle dans IPv4 et obligatoire dans IPv6.

Une base de données de sécurité, appelée SAD (Security Association Database), regroupe les caractéristiques des associations par l'intermédiaire de paramètres de la communication. L'utilisation de ces paramètres est définie dans une autre base de données, la SPD (Security Policy Database). Une entrée de la base SPD regroupe les adresses IP de la source et de la destination, ainsi que l'identité de l'utilisateur, le niveau de sécurité requis, l'identification des protocoles de sécurité mis en œuvre, etc.

Le format des paquets IPsec est illustré à la figure 34.2. La partie la plus haute de la figure correspond au format d'un paquet IP dans lequel est encapsulé un paquet TCP. La partie du milieu illustre le paquet IPsec. On voit que l'en-tête IPsec vient se mettre entre l'en-tête IP et l'en-tête TCP. La partie basse de la figure montre le format d'un paquet

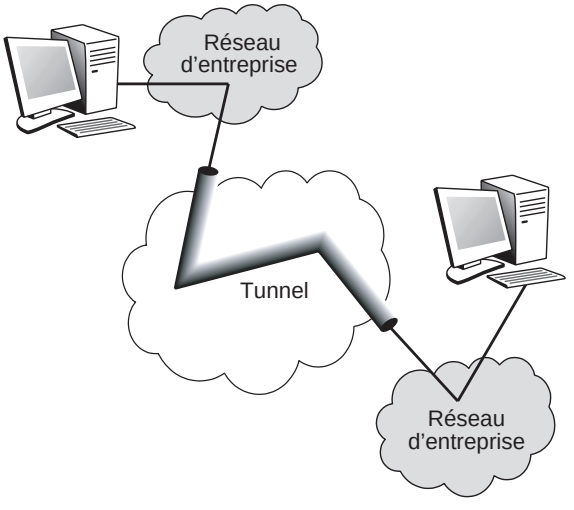
dans un tunnel IPsec. La partie intérieure correspond à un paquet IP encapsulé dans un paquet IPsec de telle sorte que le paquet IP intérieur soit bien protégé.

Figure 34.2
Format des paquets IPsec



Dans un tunnel IPsec, tous les paquets IP d'un flot sont transportés de façon totalement chiffrée. Il est de la sorte impossible de voir les adresses IP ni même les valeurs du champ de supervision du paquet IP encapsulé. La figure 34.3 illustre un tunnel IPsec.

Figure 34.3
Tunnel IPsec



L'en-tête d'authentification

L'en-tête d'authentification est ajouté immédiatement derrière l'en-tête IP standard. À l'intérieur de l'en-tête IP, le champ indiquant le prochain protocole inclus dans le paquet IP (champ Next-Header) prend la valeur 51. Cette valeur précise que les champs IPsec et d'authentification sont mis en œuvre dans le paquet IP. L'en-tête IPsec possède lui-même un champ indiquant le protocole encapsulé dans le paquet IPsec. En d'autres termes, lorsqu'un paquet IP doit être sécurisé par IPsec, il repousse la valeur de l'en-tête suivant, qui était dans le paquet IP, dans le champ en-tête suivant de la zone d'authentification d'IPsec et met la valeur 51 dans l'en-tête de départ.

La figure 34.4 présente le détail l'en-tête d'authentification. Comme indiqué précédemment, cet en-tête commence par la valeur indiquant le protocole transporté. Le champ LG (Length), sur un octet, indique la taille de l'en-tête d'authentification. Vient ensuite une zone réservée, sur 2 octets, qui prend place avant le champ sur 4 octets, donnant un index des paramètres de sécurité, qui décrit le schéma de sécurité adopté pour la communication.

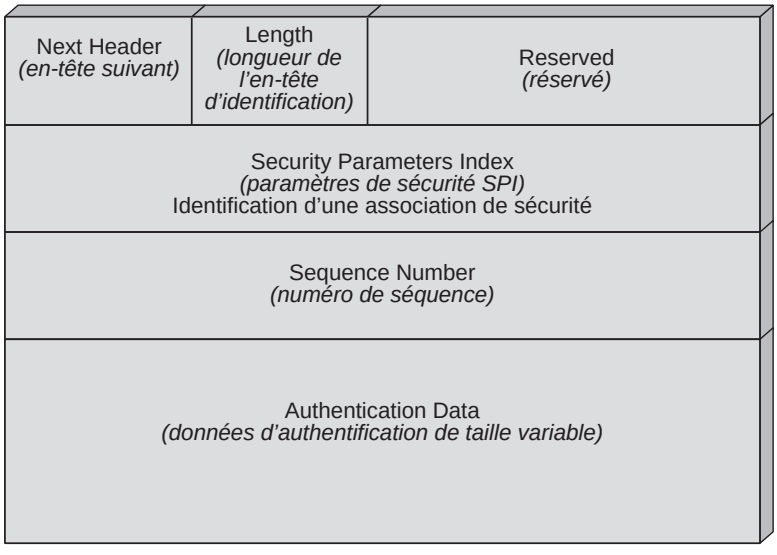


Figure 34.4
Format de l'en-tête d'authentification

Le champ numéro de séquence, qui contient un numéro de séquence unique, est nécessaire pour éviter les attaques de type rejeu, dans lesquelles le pirate rejoue exactement la même séquence de messages que l'utilisateur par une copie pure et simple. Par exemple, si vous consultez votre compte en banque et qu'un pirate recopie vos messages, même chiffrés, c'est-à-dire sans les comprendre, il peut, à la fin de votre session, rejouer la même succession de messages, qui lui ouvrira les portes de votre compte.

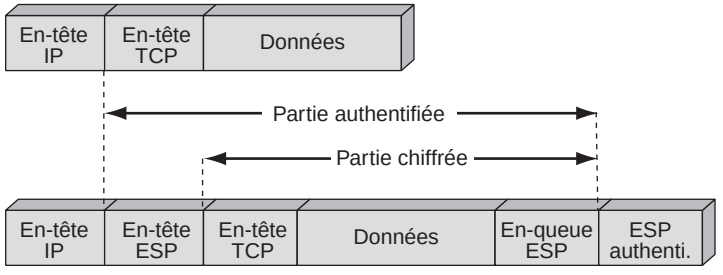
L'en-tête d'authentification se termine par les données associées à ce schéma de sécurité. Il transporte le type d'algorithme de sécurité, les clés utilisées, la durée de vie de l'algorithme et des clés, une liste des adresses IP des émetteurs qui peuvent utiliser le schéma de sécurité, etc.

L'en-tête d'encapsulation de sécurité

Pour permettre une confidentialité des données, tout en garantissant une authentification, IPsec utilise une encapsulation dite ESP (Encapsulating Security Payload), c'est-à-dire une encapsulation de la charge utile de façon sécurisée. La valeur 50 est transportée dans le champ en-tête suivant (Next-Header) du paquet IP pour indiquer cette encapsulation ESP.

La figure 34.5 illustre ce processus d'encapsulation. On s'aperçoit que l'encapsulation ESP ajoute trois champs supplémentaires au paquet IPsec : l'en-tête ESP, qui suit l'en-tête IP de départ et porte la valeur 50, le Trailer, ou en-queue, ESP, qui est chiffré avec la charge utile, et le champ d'authentification ESP de taille variable, qui suit la partie chiffrée sans être lui-même chiffré.

Figure 34.5
Processus d'encapsulation ESP



Le paquet ESP est repris à la figure 34.6 de façon un peu plus détaillée en ce qui concerne les champs internes, à partir du champ ESP d'en-tête.

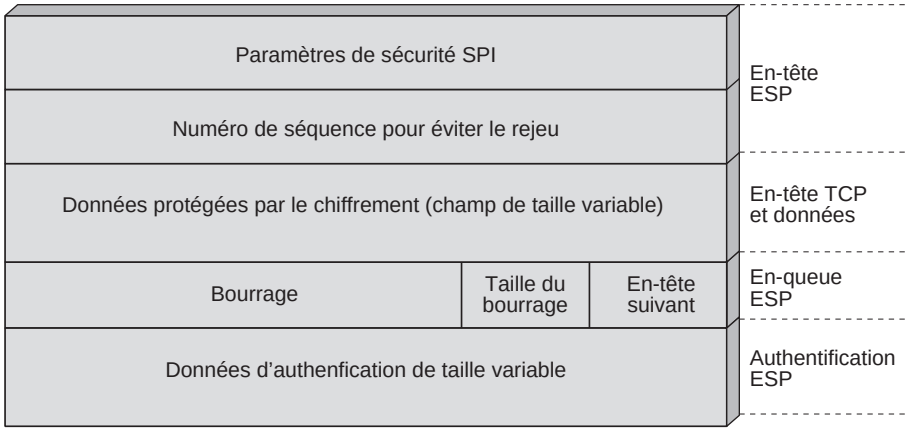


Figure 34.6
Format de l'en-tête ESP

La première partie de l'encapsulation reprend les paramètres SPI (Security Parameter Index) et numéro de séquence que nous avons déjà décrits dans l'en-tête d'authentification. Vient ensuite la partie transportée et chiffrée. L'en-queue ESP comporte une zone de bourrage optionnelle, allant de 0 à 255 octets, puis un champ longueur du bourrage (Length) et la valeur d'un en-tête suivant.

La zone de bourrage a plusieurs raisons d'être. La première provient de l'adoption d'algorithmes de chiffrement, qui exigent la présence d'un nombre de 0 déterminé après la zone chiffrée. La deuxième raison vient de la place de l'en-tête suivant, qui doit être

aligné à droite, c'est-à-dire prendre une place en fin d'un mot de 4 octets. La dernière raison est que, pour contrer une attaque, il peut être intéressant d'ajouter de l'information sans signification susceptible de leurrer un pirate.

Les compléments d'IPsec

Dans IPsec, le chiffrement ne s'effectue pas sur l'ensemble des champs, car certains champs, que l'on appelle *mutable*, changent de valeur à la traversée des routeurs, comme le champ TTL (durée de vie). Dans le calcul du champ d'authentification, le processus ne tient pas compte de ces champs mutables.

Les algorithmes de sécurité qui peuvent être utilisés dans le cadre d'IPsec sont déterminés par un certain nombre de RFC :

- Pour l'en-tête d'authentification :
  - HMAC avec MD5 : RFC 2403 ;
  - HMAC avec SHA-1 : RFC 2403.
- Pour l'en-tête ESP :
  - DES en mode CBC : RFC 2405 ;
  - HMAC avec MD5 : RFC 2403 ;
  - HMAC avec SHA-1 : RFC 2404.

La sécurité dans IPv6

Le protocole IPv6 contient les mêmes fonctionnalités qu'IPsec. On peut donc dire qu'il n'existe pas d'équivalent d'IPsec dans le contexte de la nouvelle génération IP.

Les champs de sécurité sont optionnels. Leur existence est détectée par les valeurs 50 et 51 du champ en-tête suivant (Next-Header). Globalement, la sécurité offerte par IPv6 est donc exactement la même que celle offerte par IPsec. Elle est toutefois plus simple à mettre en œuvre puisque le protocole de sécurité est dans le protocole IPv6 lui-même. On peut en déduire que la sécurisation des communications sera beaucoup plus simple avec la nouvelle génération de réseau qui utilisera IPv6.

Cela pose toutefois d'autres problèmes. Si tous les flux sont chiffrés, par exemple, il n'y a plus moyen de reconnaître les numéros de port ou les adresses source et destination, et les applications deviennent transparentes. Toutes les appliances intermédiaires, comme les pare-feu ou les contrôleurs de qualité de service, deviennent inutilisables. L'information sur le type d'application véhiculé ne se trouve qu'à la source ou à la destination, et c'est là qu'il faut venir la rechercher. De nouvelles architectures de contrôle sont donc à prévoir avec l'arrivée d'IPv6, ce qui constitue une raison de repousser cette arrivée dans beaucoup d'entreprises.

SSL (Secure Sockets Layer)

SSL est un logiciel permettant de sécuriser les communications sous HTTP ou FTP. Ce logiciel a été développé par Netscape pour son navigateur et les serveurs Web.

Le rôle de SSL est de chiffrer les messages entre un navigateur et le serveur Web interrogé. Le niveau d'architecture où se place SSL est illustré à la figure 34.7. Il s'agit d'un niveau compris entre TCP et les applicatifs.

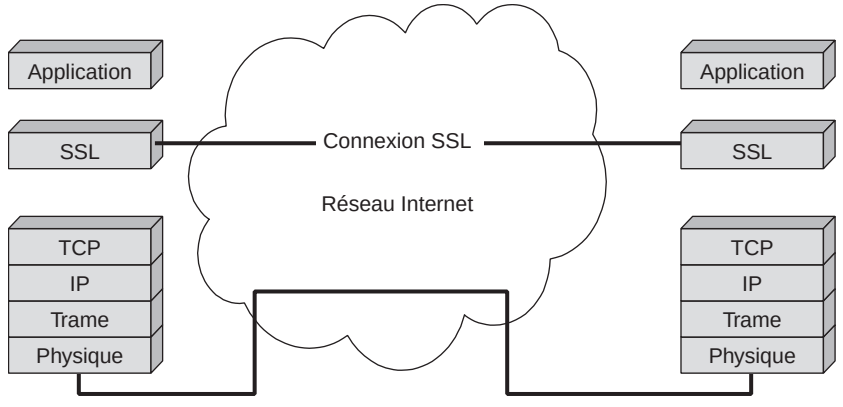


Figure 34.7
Architecture SSL

Les signatures électroniques sont utilisées pour l'authentification des deux extrémités de la communication et l'intégrité des données.

L'initialisation d'une communication SSL commence par un handshake, c'est-à-dire une poignée de main, qui permet l'authentification réciproque grâce à un tiers de confiance. La communication se continue par une négociation du niveau de sécurité à mettre en œuvre et peut se dérouler avec un chiffrement associé au niveau négocié à la phase précédente.

Les inconvénients du protocole SSL proviennent de l'utilisation d'un tiers de confiance et de la nécessité d'ouvrir le port associé à SSL dans les pare-feu. Nous verrons ultérieurement dans ce chapitre la signification exacte de l'expression « ouvrir un port ».

Le protocole SSL a vu son champ d'action dépasser la simple sécurisation d'une communication Web. Il est notamment utilisé dans le commerce électronique pour sécuriser la transmission du numéro de carte de crédit. Un autre protocole, S-HTTP (Secure HTTP), assez semblable à SSL, a été développé pour sécuriser les communications sous HTTP, mais il est beaucoup moins utilisé.

L'authentification de SSL s'appuie sur la cryptographie asymétrique par le biais de certificats à clé publique. Un client peut s'authentifier automatiquement auprès d'un serveur SSL utilisant la paire de clés publiques nécessaire. SSL peut utiliser différents mécanismes de chiffrement. En règle générale, la négociation entre le client et le serveur SSL permet de définir le meilleur algorithme commun. Classiquement, un serveur SSL utilise une clé publique RSA pour établir un ensemble de clés secrètes partagées utilisées avec un algorithme de chiffrement RC4 pour le chiffrement des données et MD5 pour l'intégrité.

Le protocole SSLv3 propose une architecture plus évoluée, qui contient un générateur de clés, des fonctions de hachage et des algorithmes de chiffrement et de gestion de certificats. Cette architecture est représentée à la figure 34.8.

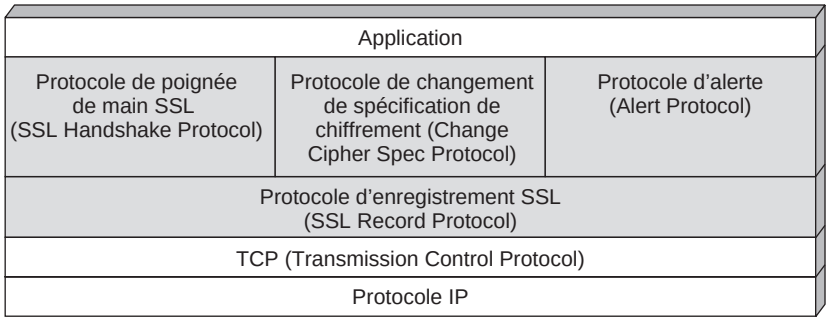


Figure 34.8

Architecture du protocole SSLv3

Le protocole de changement de spécification de chiffrement permet de modifier l'algorithme de chiffrement en cours de communication de sorte à garantir la confidentialité des données transportées. Le protocole d'alerte permet d'envoyer des alertes, accompagnées de leur importance. Ces alertes peuvent être un certificat inconnu, révoqué, expiré, etc. Les alertes de haut niveau entraînent l'arrêt de la communication.

Le protocole Handshake a pour objectif d'authentifier le serveur depuis le client, de négocier la version du protocole, de sélectionner les algorithmes de chiffrement, d'utiliser des techniques de chiffrement à clé publique pour générer et distribuer des clés secrètes et d'établir des connexions SSL chiffrées.

Messages du protocole Handshake

Les messages échangés pour réaliser le protocole Handshake sont les suivants :

- CLIENTHELLO : initialisation de la communication par l'envoi d'un hello du client vers le serveur.
- SERVERHELLO : en retour du message précédent, cette réponse peut contenir un certificat et demander une authentification de la part du client.
- SERVERKEYEXCHANGE : si les certificats ne sont pas pris en charge, ce message permet d'effectuer l'échange de clés publiques.
- SERVERHELLODONE : permet d'indiquer que la partie serveur du message hello est achevée.
- CERTIFICATEREQUEST : requête envoyée par le serveur au client lui demandant de s'authentifier. Le client répond soit avec un message envoyant le certificat, soit avec une alerte indiquant qu'il ne possède pas de certificat.
- CERTIFICATEMESSAGE : message qui envoie le certificat réclamé par le serveur.
- NOCERTFICATE : message d'alerte qui indique que le client ne possède aucun certificat susceptible de correspondre à la demande du serveur.
- CLIENTKEYEXCHANGE : échange de la clé du client avec le serveur.
- FINISHED : message qui conclut le handshake pour indiquer la fin de la mise en place de la communication.

Le protocole d'enregistrement SRP (SSL Record Protocol) n'est qu'une encapsulation des protocoles situés juste au-dessus, comme le protocole Handshake.

Le protocole SSLv3.0 et son successeur TLS1.0 ne présentent que des différences mineures entre eux mais ne sont cependant pas interopérables. La principale différence entre les deux concerne les méthodes de chiffrement puisque TLS n'impose aucune restriction.

EAP (Extensible Authentication Protocol)

EAP est une extension de PPP dédiée à l'authentification. PPP débute par une phase d'établissement de la liaison. Cette phase est gérée par le protocole LCP (Link Control Protocol), qui configure et négocie l'authentification. La deuxième phase est optionnelle et concerne l'authentification. Pour réaliser l'authentification, PPP définit une liste de protocoles d'authentification, dont EAP est le plus classique et le plus utilisé.

EAP est un protocole d'authentification général, qui supporte de multiples méthodes d'authentification, telles que Kerberos, TLS, MS-Chap, SIM, etc. De nouveaux mécanismes d'authentification peuvent ainsi être définis au-dessus d'EAP. Seul le champ type du paquet EAP, codé sur un octet, limite le nombre de mécanismes d'authentification. Le standard EAP a été conçu comme protocole générique pour le transport du trafic des protocoles d'authentification. Il est construit autour d'un modèle de communication demandant un défi (challenge), auquel une réponse doit être apportée pour qu'il y ait authentification.

Comme illustré à la figure 34.9, quatre types de messages EAP permettent de réaliser l'authentification d'un client sur un serveur :

- EAP REQUEST : demande d'authentification ;
- EAP RESPONSE : réponse à une requête d'authentification ;
- EAP SUCCESS : pour indiquer le succès de l'authentification ;
- EAP FAILURE : pour informer le client du résultat négatif de l'authentification.

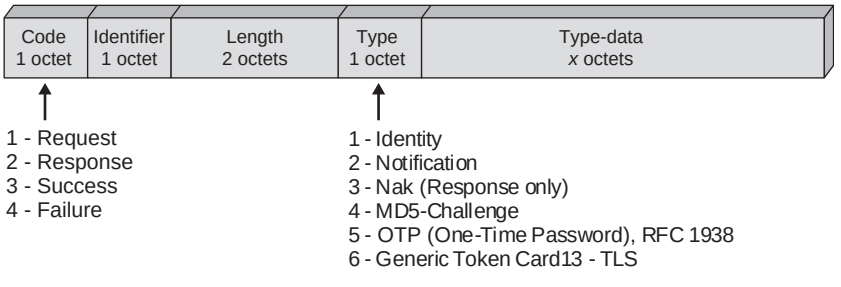


Figure 34.9
Format d'un paquet EAP

L'authentification EAP se déroule de la manière suivante :

1. Lorsque la phase d'établissement de la liaison PPP est terminée, le serveur d'authentification envoie une demande d'identité.

2. Le client envoie un paquet EAP RESPONSE dans lequel il fournit son identité et les méthodes d'authentification qu'il supporte. La phase d'authentification débute à cet instant.
3. Le serveur envoie un défi au client.
4. Le client y répond par un message EAP RESPONSE, dans lequel il envoie le défi chiffré avec sa clé secrète.
5. Le serveur met fin à la phase d'authentification par l'intermédiaire d'un paquet de succès ou d'échec.

Si la phase d'authentification s'est bien déroulée, le serveur d'authentification peut transmettre une clé de chiffrement au client, lequel l'utilisera pour chiffrer les données émises. Cette dernière phase est optionnelle pour le protocole EAP car elle dépend du protocole d'authentification utilisé.

Le protocole EAP est extensible puisque tout mécanisme d'authentification peut être encapsulé à l'intérieur des messages EAP, comme l'illustre la figure 34.10. Au niveau supérieur de la figure se trouvent les méthodes d'authentification, comme TLS, MS-Chap, SIM, etc. Vient ensuite le niveau d'encapsulation de la méthode d'authentification de la trame EAP. La trame EAP elle-même est encapsulée dans une trame de transport. Cette encapsulation peut s'effectuer soit dans une trame EAP over Radius, c'est-à-dire dans une trame RADIUS, soit dans une trame EAPoL (EAP over LAN), qui est utilisée dans les réseaux locaux, en particulier les réseaux locaux sans fil de type Wi-Fi.

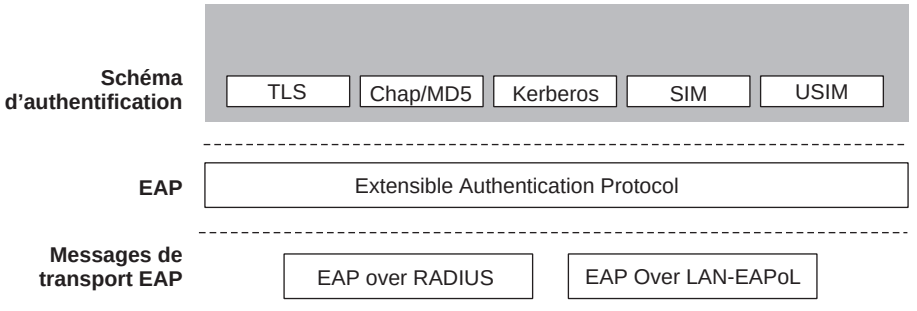


Figure 34.10
Architecture d'EAP

Un autre atout d'EAP est qu'il est conçu pour fonctionner au-dessus de la couche liaison. Il ne nécessite donc pas le niveau IP mais inclut son propre support pour la livraison et la retransmission. Développé pour être utilisé avec PPP, c'est un protocole générique qui supporte la majorité des normes de niveau liaison.

EAP-SIM

Les réseaux de mobiles de deuxième génération ainsi que les hotspots Wi-Fi peuvent utiliser l'authentification mutuelle EAP-SIM. Ce protocole propose des améliorations aux procédures d'authentification utilisées par le GSM en fournissant une authentification entre le centre d'authentification de l'opérateur mobile et chaque utilisateur qui possède une carte à puce SIM et pour laquelle les algorithmes d'authentification sont présents à la fois dans le réseau et dans toutes les cartes à puce SIM.

L'authentification EAP-SIM est illustrée à la figure 34.11. Cette solution est présentée pour un hotspot Wi-Fi dans lequel le client, appelé supplicant dans la norme, possède une carte à puce capable de gérer le protocole EAP-SIM, la carte à puce étant connectée à l'ordinateur personnel par une interface USB ou un lecteur de carte à puce. Cette solution peut être relayée vers un serveur d'authentification, ou AS (Authentication Server), d'un opérateur de mobiles. Le serveur d'authentification est généralement un serveur RADIUS. La communication traverse le point d'accès Wi-Fi, appelé authenticator, puisque c'est lui que le client voit et non pas le serveur d'authentification.

La trame EAP est d'abord transportée sur l'interface avec l'ordinateur personnel sur une interface ISO 7816 puis sur l'interface radio en EAPoL et enfin à partir du point d'accès en EAP over RADIUS.

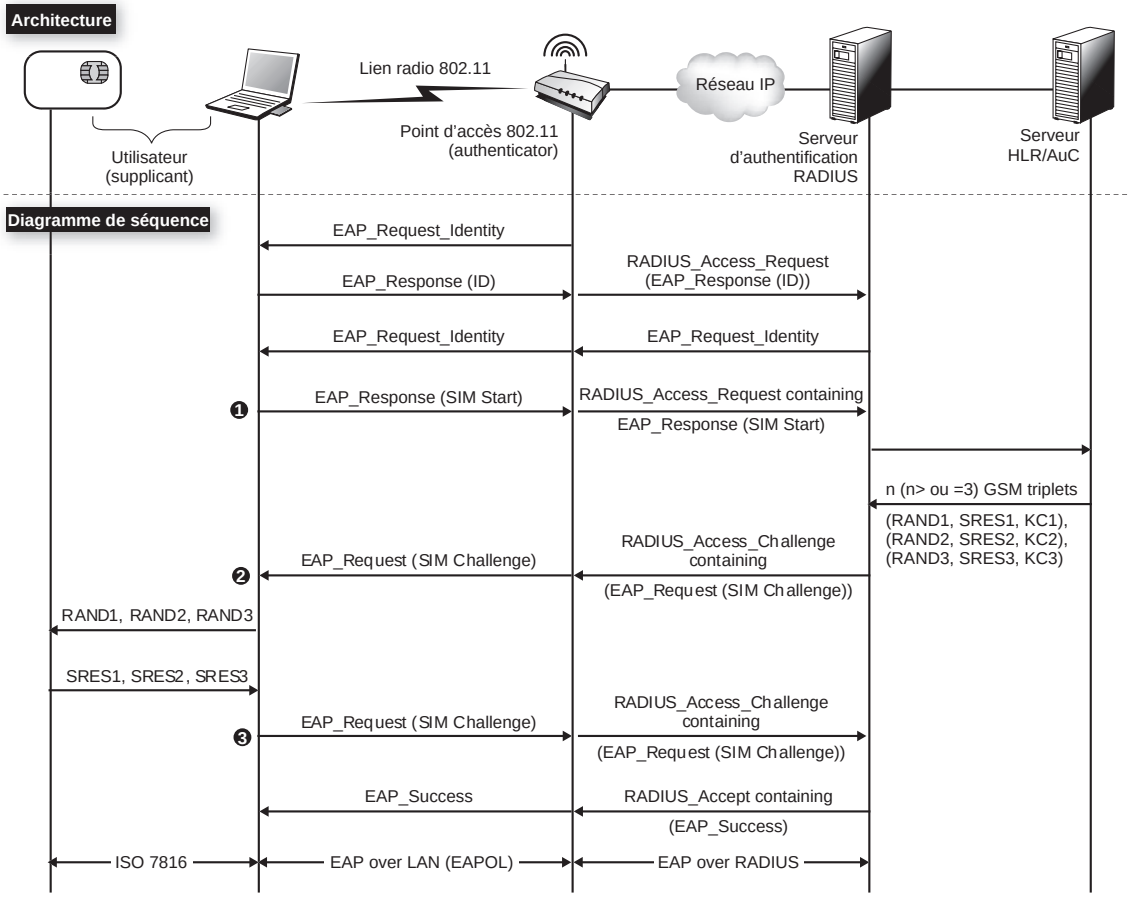


Figure 34.11
Authentification EAP-SIM

EAP-TLS

La procédure EAP-TLS devient la solution la plus courante pour l'authentification d'un client, que ce soit sur un poste de travail ou sur une carte à puce. TLS est la procédure que nous avons introduite à la section précédente comme prolongement du protocole SSLv3 et qui permet à un client et à son serveur d'échanger leur certificat numérique. Le serveur

présente un certificat au client, que ce dernier valide. Optionnellement, le client présente son certificat au serveur. Le certificat peut être protégé côté client par un mot de passe ou le code PIN d'une carte à puce.

La figure 34.12 montre les différents messages échangés lors de la phase d'authentification d'un client EAP-TLS, toujours dans le cas d'un réseau sans fil Wi-Fi et d'une carte à puce liée à l'ordinateur personnel. La figure représente une authentification réussie entre l'authentifiant (Authentication Server) RADIUS et le client (supplicant) avec comme intermédiaire le point d'accès (authenticator). Une communication EAP-TLS commence par la négociation EAP entre le client, demandant un accès au réseau, et le point d'accès :

1. Le point d'accès envoie un paquet EAP REQUEST IDENTITY.
2. Le client répond par un paquet EAP RESPONSE IDENTITY contenant l'identité de l'utilisateur.
3. Le serveur envoie un paquet EAP TLS START.
4. La réponse du client est un paquet EAP RESPONSE contenant un message TLS CLIENT HELLO HANDSHAKE. Le message CLIENT HELLO contient la version TLS du client, ainsi qu'un nombre aléatoire et une liste des algorithmes de chiffrement supportés par le client.
5. Le serveur envoie un paquet EAP REQUEST dont les données contiennent un message SERVER HELLO HANDSHAKE. Ce message spécifie la version de TLS du serveur, ainsi qu'un autre nombre aléatoire, un identifiant de session et un message CIPHERSUITE, qui correspond à l'algorithme de chiffrement choisi.
6. Le client répond par un paquet EAP RESPONSE dont le champ de données encapsule un message TLS CHANGE CIPHER SPEC et un message FINISHED HANDSHAKE.

TTLS et PEAP

La sélection d'une méthode d'authentification est une décision importante pour le déploiement sécurisé d'un réseau. La méthode d'authentification conduit au choix du serveur d'authentification, qui, à son tour, conduit au choix du logiciel client.

Dans le cas où une infrastructure PKI n'est pas déjà déployée, il existe des solutions de rechange à EAP-TLS pour éviter la complexité du certificat client. Ces méthodes d'authentification présentent un niveau de sécurité équivalent à celui obtenu avec les certificats numériques et permettent de s'affranchir des barrières liées à la mise en place d'une infrastructure PKI. Par exemple, TTLS (Tunneled Transport Layer Security) et PEAP (Protected EAP) conservent tous deux les fortes fondations de chiffrement de TLS mais utilisent d'autres mécanismes pour authentifier le client.

Ces protocoles commencent par établir un canal sécurisé, ou tunnel TLS, après quoi le client authentifie le serveur. Dans une seconde étape, des messages d'authentification sont échangés :

- TTLS échange des AVP (Attribute-Value Pair) avec un serveur qui les valide pour tout type d'authentification.
- PEAP utilise le canal TLS pour protéger un second échange EAP. MS-Chap peut être utilisé pour les clients n'ayant pas de PKI. Pour les clients ayant une PKI, EAP-TLS peut être utilisé. L'avantage par rapport au protocole EAP-TLS classique est que l'identité du client est protégée lors de l'échange.

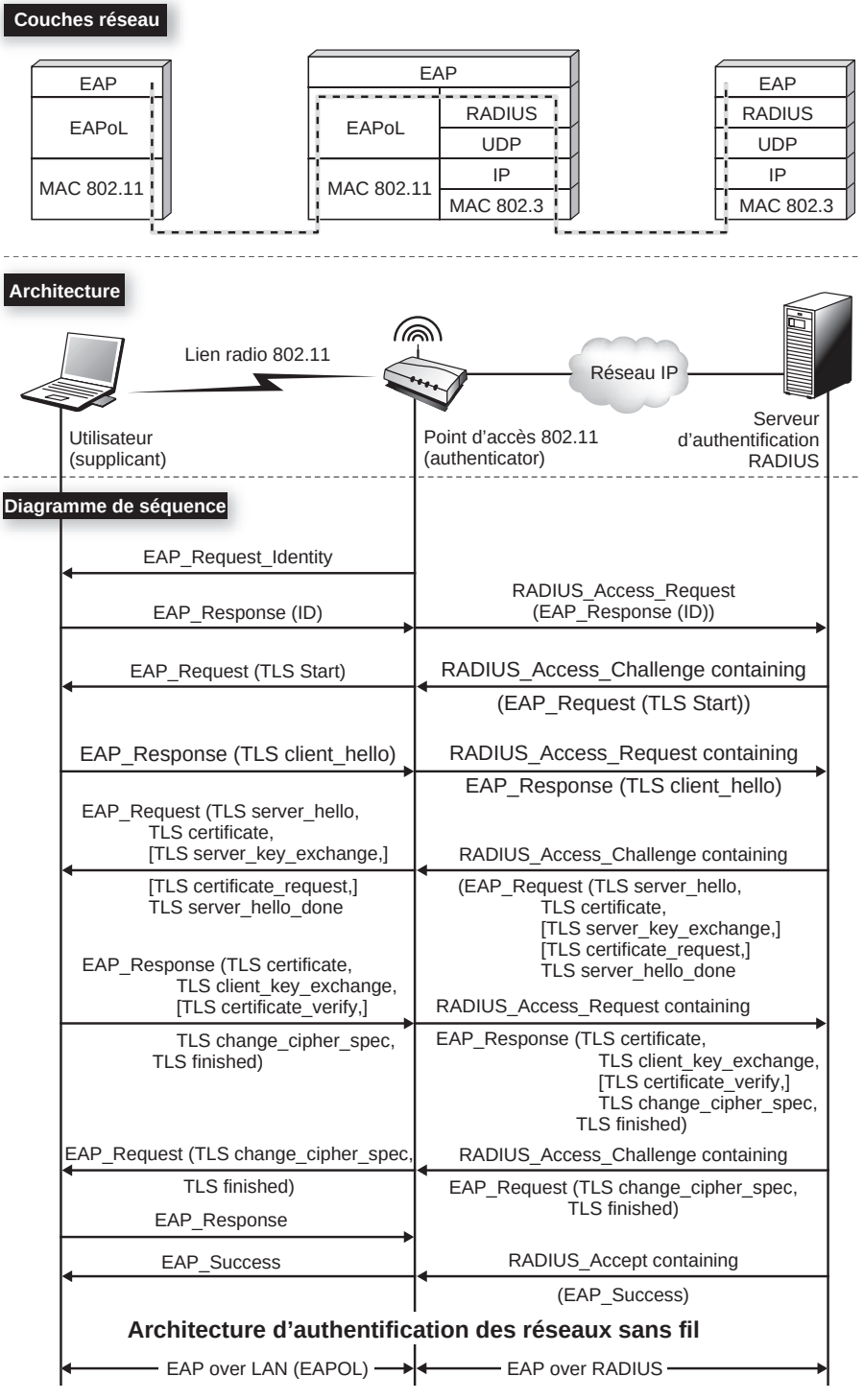


Figure 34.12
Authentification EAP-TLS

RADIUS (Remote Authentication Dial-In User Server)

Quel que soit le choix du mécanisme d'authentification entre le point d'accès et le serveur d'authentification, les paquets EAP sont généralement acheminés grâce au protocole RADIUS. RADIUS est depuis longtemps le protocole AAA (Authentication, Authorization, Accounting) le plus largement adopté. Utilisé par les ISP pour authentifier les utilisateurs, il est principalement conçu pour transporter des données d'authentification, d'autorisation et de facturation entre des NAS (Network Access Server) distribués, qui désirent authentifier leurs utilisateurs et un serveur d'authentification partagé.

RADIUS utilise une architecture client-serveur qui repose sur le protocole UDP. Les NAS, qui jouent le rôle de client, sont responsables du transfert des informations envoyées par l'utilisateur vers les serveurs RADIUS. Ces derniers prennent en charge la réception des demandes d'authentification, l'authentification des utilisateurs et les réponses contenant toutes les informations de configuration nécessaires aux NAS. Les serveurs RADIUS peuvent également agir comme proxy pour d'autres serveurs RADIUS.

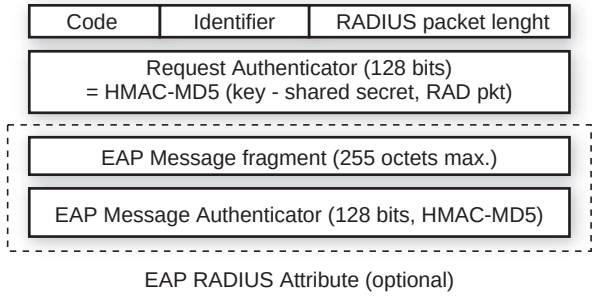
Si un équipement mobile a besoin d'accéder au réseau en utilisant RADIUS pour l'authentification, il doit présenter au NAS des crédits d'authentification (identifiant utilisateur, mot de passe, etc.). Ce dernier les transmet au serveur RADIUS en lui envoyant un ACCESS-REQUEST. Le NAS et les proxy RADIUS ne peuvent interpréter ces crédits d'authentification car ces derniers sont chiffrés entre l'utilisateur et le serveur RADIUS destinataire. À réception de cette requête, le serveur RADIUS vérifie l'identifiant du NAS puis les crédits d'authentification de l'utilisateur dans une base de données LDAP (Lightweight Directory Access Protocol) ou autre.

Les données d'autorisation échangées entre le client (le NAS) et le serveur RADIUS sont toujours accompagnées d'un secret partagé. Ce secret est utilisé pour vérifier l'authenticité et l'intégrité de chaque paquet entre le NAS et le serveur.

La figure 34.13 illustre le format type d'un paquet RADIUS. L'authentifiant du message sur 128 bits n'est autre qu'un résumé HMAC-MD5 du paquet échangé, calculé à l'aide du secret partagé.

Figure 34.13

Format type d'un paquet RADIUS



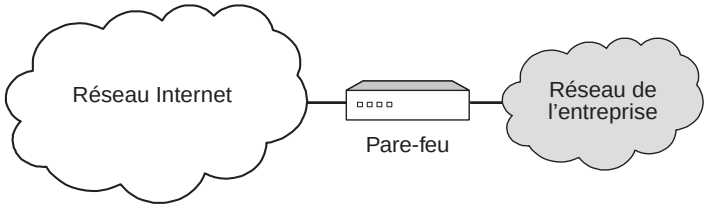
RADIUS peut supporter plusieurs mécanismes d'authentification. Il peut utiliser, par exemple, des procédures de défi/réponse (Chap) et des messages ACCEPT-CHALLENGE.

L'authentification par mot de passe, ou PAP (Password Authentication Protocol), est aussi prise en charge. Les serveurs RADIUS répondent aux demandes d'authentification par des messages ACCESS-ACCEPT ou ACCESS-REJECT. Les paquets ACCESS-ACCEPT fournissent les informations de configuration nécessaires pour autoriser les clients RADIUS à commencer une connexion sécurisée avec des utilisateurs.

Les pare-feu

Un pare-feu est un équipement de réseau, la plupart du temps de type routeur, placé à l'entrée d'une entreprise afin d'empêcher l'entrée ou la sortie de paquets non autorisés par l'entreprise. La situation géographique d'un pare-feu est illustrée à la figure 34.14.

Figure 34.14
Situation d'un pare-feu dans l'entreprise



Toute la question est de savoir comment reconnaître les paquets à accepter et à refuser. Il est possible de travailler de deux façons :

- interdire tous les paquets sauf ceux d'une liste prédéterminée ;
- accepter tous les paquets sauf ceux d'une liste prédéterminée.

En règle générale, un pare-feu utilise la première solution en interdisant tous les paquets, sauf ceux qu'il est possible d'authentifier par rapport à une liste de paquets que l'on souhaite laisser entrer. Cela comporte toutefois un inconvénient : lorsqu'un client de l'entreprise se connecte sur un serveur à l'extérieur, la sortie par le pare-feu est acceptée puisque authentifiée. La réponse est généralement refusée, puisque le port sur lequel elle se présente n'a aucune raison d'accepter ce message s'il est bloqué par mesure de sécurité. Pour que la réponse soit acceptée, il faudrait que le serveur puisse s'authentifier et que le pare-feu lui permette d'accéder au port concerné.

L'autre option est évidemment beaucoup plus dangereuse puisque tous les ports sont ouverts sauf ceux qui ont été bloqués. Une attaque ne se trouve pas bloquée tant qu'elle n'utilise pas les accès interdits.

Avant d'aller plus loin, considérons les moyens d'accepter ou de refuser des flots de paquets. Les filtres permettent de reconnaître un certain nombre de caractéristiques des paquets, comme l'adresse IP d'émission, l'adresse IP de réception, parfois les adresses de niveau trame, le numéro de port et plus généralement tous les éléments disponibles dans l'en-tête du paquet IP. Pour ce qui concerne la reconnaissance de l'application, les filtres sont essentiellement réalisés sur les numéros de port utilisés par les applications. Nous verrons toutefois un peu plus loin que cette solution n'est pas imparable.

Un numéro de port est en fait une partie d'un numéro de socket, ce dernier étant, comme expliqué au chapitre précédent, la concaténation d'une adresse IP et d'un numéro de port. Les numéros de port correspondent à des applications. Les principaux ports sont recensés au tableau 34.1.

Quelques ports réservés TCP		
N° de port	**Service**	**Rôle**
1	tcpmux	Multiplexeur de service TCP
3	compressnet	Utilitaire de compression
7	echo	Fonction écho
9	discard	Fonction d'élimination
11	users	Utilisateurs
13	daytime	Jour et heure
15	netstat	État du réseau
20	ftp-data	Données du protocole FTP
21	ftp	Protocole FTP
23	telnet	Protocole Telnet
25	smtp	Protocole SMTP
37	heure	Serveur heure
42	name	Serveur nom d'hôte
43	whols	Nom NIC
53	domain	Serveur DNS
77	rje	Protocole RJE
79	finger	Finger
80	http	Service WWW
87	link	Liaison TTY
103	X400	Messagerie X.400
109	pop	Protocole POP
144	news	Service News
158	tcprepo	Répertoire TCP
Quelques ports réservés UDP		
7	echo	Service écho
9	rejet	Service de rejet
53	dsn	Serveur de nom de domaine
67	dhcp	Serveur de configuration DHCP
68	dhcp	Client de configuration DHCP

TABLEAU 34.1 • Principaux ports TCP et UDP

Un pare-feu contient donc une table, qui indique les numéros de port acceptés.

Le tableau 34.2 donne la composition d'un pare-feu classique, dans lequel seulement six ports sont ouverts, dont l'un ne l'est que pour une adresse de réseau de classe C spécifique.

Port accepté	Adresse IP
21	*
23	*
25	Adresse réseau C – adresse réseau B
43	*
69	*
79	*

TABLEAU 34.2 • Composition d'un pare-feu classique

Les pare-feu peuvent être de deux types, proxy et applicatif. Dans le premier cas, le pare-feu a pour objectif de couper la communication entre un client et un serveur ou entre un client et un autre client. Ce type de pare-feu ne permet pas à un attaquant d'accéder directement à la machine attaquée, ce qui donne une forte protection supplémentaire. Dans le second cas, le pare-feu détecte les flots applicatifs et les interrompt ou non suivant les éléments filtrés. Dans tous les cas, il faut utiliser des filtres plus ou moins puissants.

Les filtres

Comme expliqué précédemment, les filtres sont essentiellement appliqués sur les numéros de port. La gestion de ces numéros de port n'est toutefois pas simple. En effet, de plus en plus de ports sont dynamiques. Avec ces ports, l'émetteur envoie une demande sur le port standard, mais le récepteur choisit un nouveau port disponible pour effectuer la communication. Par exemple, l'application RPC (Remote Procedure Call) affecte dynamiquement les numéros de port. La plupart des applications P2P (Peer-to-Peer) ou de signalisation de la téléphonie sont également dynamiques.

L'affectation dynamique de port peut être contrôlée par un pare-feu qui se comporte astucieusement. La communication peut ainsi être suivie à la trace, et il est possible de découvrir la nouvelle valeur du port lors du retour de la demande de transmission d'un message TCP. À l'arrivée de la réponse indiquant le nouveau port, il faut détecter le numéro du port qui remplace le port standard. Un cas beaucoup plus complexe est possible, dans lequel l'émetteur et le récepteur se mettent directement d'accord sur un numéro de port. Dans ce cas, le pare-feu ne peut détecter la communication, sauf si tous les ports sont bloqués. C'est la raison essentielle pour laquelle les pare-feu n'acceptent que des communications déterminées à l'avance.

Cette solution de filtrage et de reconnaissance des ports dynamiques n'est toutefois pas suffisante, car il est toujours possible pour un pirate de transporter ses propres

données à l'intérieur d'une application standard sur un port ouvert. Par exemple, un tunnel peut être réalisé sur le port 80, qui gère le protocole HTTP. À l'intérieur de l'application HTTP, un flot de paquets d'une autre application peut passer. Le pare-feu voit entrer une application HTTP, qui, en réalité, délivre des paquets d'une autre application.

Une entreprise ne peut pas bloquer tous les ports, sans quoi ses applications ne pourraient plus se dérouler. On peut bien sûr essayer d'ajouter d'autres facteurs de détection, comme l'appartenance à des groupes d'adresses IP connues, c'est-à-dire à des ensembles d'adresses IP qui ont été définies à l'avance. De nouveau, l'emprunt d'une adresse connue est assez facile à mettre en œuvre. De plus, les attaques les plus dangereuses s'effectuent par des ports qu'il est impossible de bloquer, comme le port DNS. Une des attaques les plus dangereuses s'effectue par un tunnel sur le port DNS. Encore faut-il que la machine réseau de l'entreprise qui gère le DNS ait des faiblesses pour que le tunnel puisse se terminer et que l'application pirate s'exprime dans l'entreprise. Nous verrons à la section suivante comment il est possible de renforcer la sécurité des pare-feu.

Pour sécuriser l'accès à un réseau d'entreprise, une solution beaucoup plus puissante consiste à filtrer non plus aux niveaux 3 ou 4 (adresse IP ou adresse de port) mais au niveau applicatif. Cela s'appelle un filtre applicatif. L'idée est de reconnaître directement sur le flot de paquets l'identité de l'application plutôt que de se fier à des numéros de port. Cette solution permet d'identifier une application insérée dans une autre et de reconnaître les applications sur des ports non conformes. La difficulté avec ce type de filtre réside dans la mise à jour des filtres chaque fois qu'une nouvelle application apparaît. Le pare-feu muni d'un tel filtre applicatif peut toutefois interdire toute application non reconnue, ce qui permet de rester à un niveau de sécurité élevé.

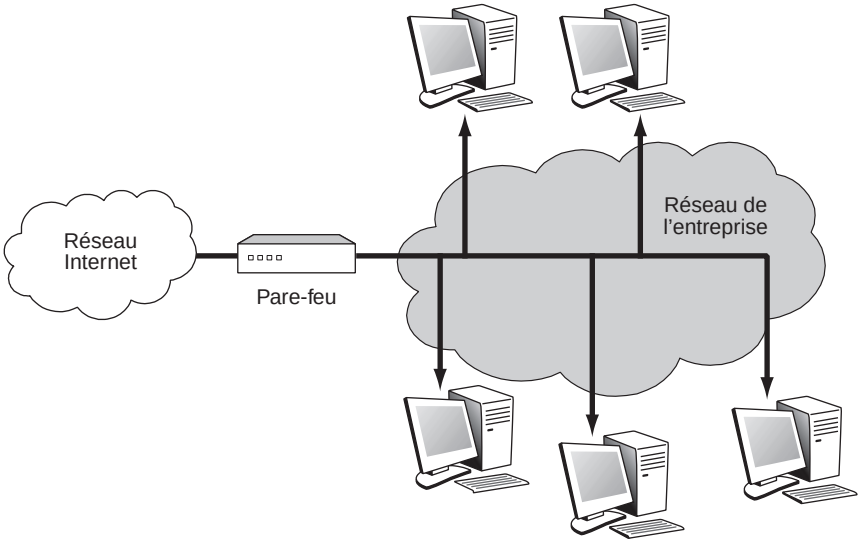
La sécurité autour du pare-feu

Comme nous l'avons vu, le pare-feu vise à filtrer les flots de paquets sans empêcher le passage des flots utiles à l'entreprise, flots que peut essayer d'utiliser un pirate. La structure de l'entreprise peut être conçue de différentes façons. Deux solutions générales sont mises en œuvre. La première est illustrée à la figure 34.15, et la seconde à la figure 34.16.

Dans le premier cas, la communication, après avoir traversé le pare-feu, se dirige au travers du réseau d'entreprise vers le poste de travail de l'utilisateur. Dans ce cas, il faut que les postes de travail de l'utilisateur soient des machines sécurisées afin d'empêcher les flots pirates qui auraient réussi à passer le pare-feu d'entrer dans des failles du système de la station. Comme cette solution est très difficile à sécuriser, puisqu'elle dépend de l'ensemble des utilisateurs d'une entreprise, la plupart des architectes réseau préfèrent mettre en entrée de réseau une machine sécurisée, que l'on appelle machine bastion *(voir figure 34.16).*

La machine bastion apporte quelques difficultés supplémentaires de gestion. En effet, elle prend en charge l'ouverture et la fermeture des communications d'un utilisateur avec l'extérieur. Par exemple, un client avec son navigateur ne peut plus accéder à un serveur

Figure 34.15
Place d'un pare-feu dans l'infrastructure réseau



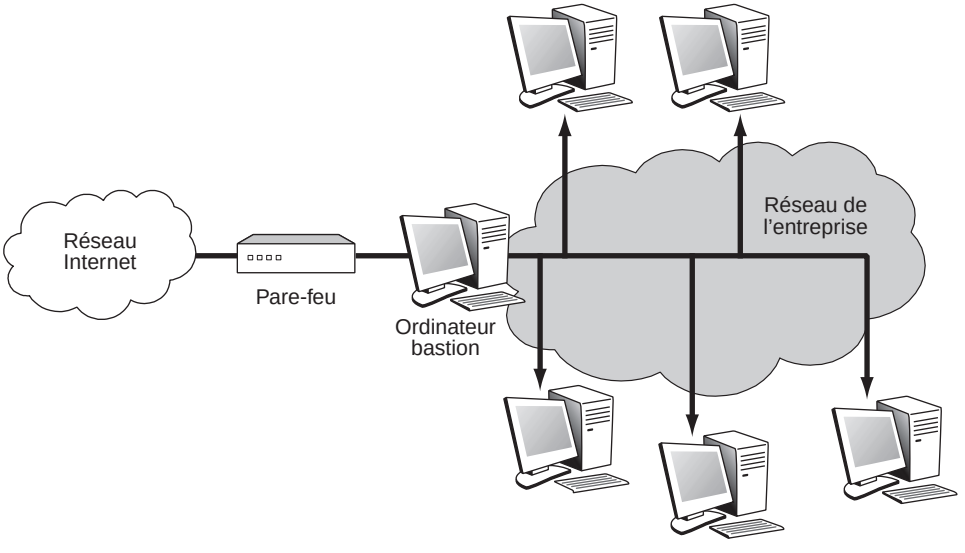


Figure 34.16
Pare-feu associé à une machine bastion

externe puisque la machine bastion l'arrête automatiquement. Le bastion doit être équipé d'un serveur proxy, et chaque navigateur être configuré pour utiliser le proxy. La communication se fait donc en deux temps. L'utilisateur communique avec son proxy, et celui-ci ouvre une communication avec le serveur distant. Lorsqu'une page parvient au proxy, ce dernier peut la distribuer au client. Le bastion peut d'ailleurs servir de cache pour les pages standards utilisées par une entreprise.

Le défaut de cette dernière architecture provient de sa relative lourdeur, puisqu'il est demandé à une machine spécifique d'effectuer le travail réseau pour toutes les machines de l'entreprise. De plus, la sécurité de toute l'entreprise peut être menacée si l'ordinateur

bastion n'est pas parfaitement sécurisé, car un pirate externe peut avoir accès à l'ensemble des ressources de l'entreprise. De fait, l'architecture de sécurité peut s'avérer plus complexe lorsqu'un ordinateur bastion est mis en place.

La figure 34.17 illustre quelques-unes des architectures de sécurité qui peuvent être mises en place.

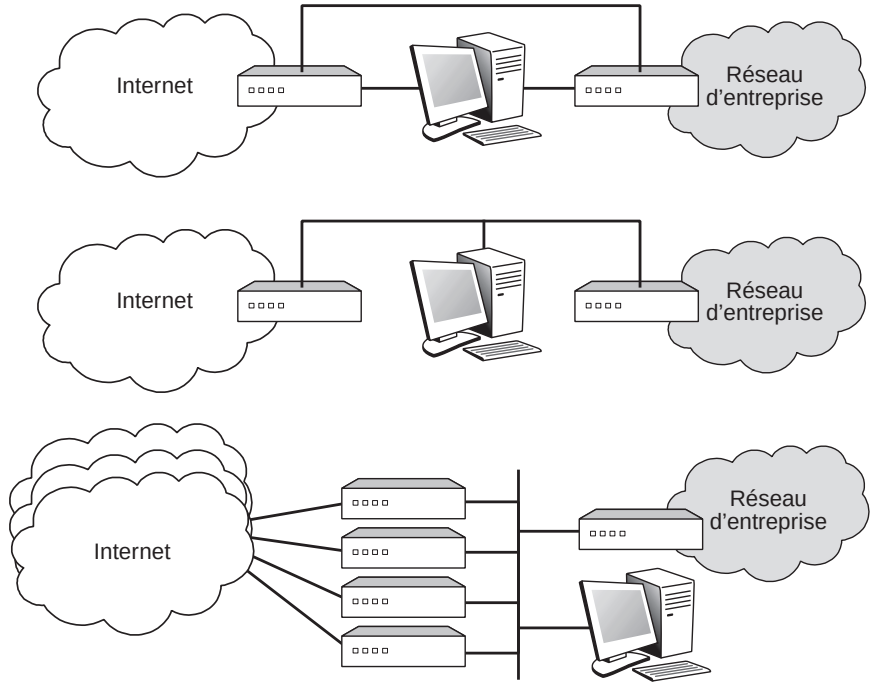


Figure 34.17
Architectures de sécurité avec machine bastion

La partie supérieure de la figure représente une organisation assez classique, dans laquelle l'ordinateur bastion est protégé des deux côtés par des pare-feu, pour filtrer aussi bien ce qui arrive de l'entreprise que ce qui arrive de l'extérieur. Le schéma montre deux pare-feu. Il est possible d'utiliser un seul pare-feu connecté à l'ordinateur bastion. Il est aussi possible de mettre en place manuellement une connexion directe entre les deux pare-feu pour effectuer des tests et des mises au point.

La deuxième partie de la figure est assez semblable à la précédente. Elle montre toutefois une organisation un peu différente, utilisant un réseau local pour relier les deux pare-feu et l'ordinateur bastion. La troisième partie de la figure montre une architecture encore plus complexe, dans laquelle une entreprise peut accéder à plusieurs opérateurs simultanément. Dans ce cas, un pirate peut entrer dans le réseau d'un opérateur en provenance d'un autre opérateur en passant par la passerelle d'une entreprise. Là, le piratage ne vise pas l'entreprise mais une autre entreprise, située sur le réseau de l'opérateur piraté. Pour sécuriser ce passage, l'ordinateur bastion doit de nouveau jouer le rôle de proxy, empêchant le passage direct.

Les virus

Les virus sont des programmes, généralement écrits en langage machine, susceptibles de s'introduire dans un ordinateur et de s'y exécuter. L'exécution peut produire de nombreux effets, allant du blocage d'une fonction à la destruction des ressources de l'ordinateur, comme l'effacement de la mémoire ou du disque dur, en passant par l'émission de messages incontrôlés.

Les logiciels antivirus ont pour fonction de détecter la présence de virus sur une machine et de les détruire. Cependant, comme nous allons le voir, certains virus sont résistants, et les logiciels antivirus peuvent avoir du mal à les détecter.

On dénombre un très grand nombre de techniques de virus, notamment les suivantes :

- *Boot sector virus,* ou virus travaillant sur le programme de démarrage. Ce programme se met en route au moment de la mise en marche de la station terminale. Le virus se trouve sur le disque dur et peut se dupliquer sur les disquettes ou les CD. Suivant son origine, le virus bloque un certain nombre de fonctions, parfois de façon aléatoire afin de ne pas se faire détecter. Il peut aussi empêcher tout démarrage de la machine en ne permettant pas à une instruction importante du programme de démarrage de se dérouler.

- *File infected virus.* Ce sont les plus courants. Ils s'attachent à un programme exécutable particulier et, en s'exécutant, bloquent la mise en route du programme, tout en s'attachant à d'autres programmes.

- *Polymorphic virus.* Le rôle de ces virus est de ne pas se faire détecter, tout en causant un certain nombre d'ennuis à l'utilisateur. Ils se modifient en passant à un autre programme, de telle sorte qu'ils sont parfois très difficiles à détecter puisque non répertoriés dans une forme spécifique à un programme.

- *Stealth virus,* que l'on peut traduire par virus furtifs. Comme les précédents, ils tentent de ne pas se faire détecter facilement tout en occasionnant des dégâts aux programmes auxquels ils s'accrochent. Une des méthodes qu'ils emploient le plus fréquemment consiste à s'incruster dans les programmes en prenant la place de quelques lignes de code de telle sorte que la taille exacte du programme reste inchangée.

- *Encrypted virus.* Ces virus forment une famille très délicate à repérer puisqu'ils sont chiffrés et que les antivirus n'ont pas la possibilité de les déchiffrer pour les détecter. Ces virus doivent pouvoir être déchiffrés pour être mis en œuvre. Ils nécessitent donc un environnement qui leur est adapté. Ils utilisent généralement les techniques de chiffrement utilisées classiquement dans les systèmes d'exploitation qu'ils attaquent.

- *Worms,* ou vers. Ces virus sont de nature différente. Ce sont eux-mêmes des programmes qui transportent des virus. Beaucoup d'attaques sur les messageries s'effectuent en attachant un vers au message. L'utilisateur à qui l'on a fait croire à l'utilité de ce programme l'ouvre et l'exécute. Le virus attaché peut alors commencer à infecter la machine.

- *Trojan horses*, ou chevaux de Troie. Ces virus bien connus sont des programmes qui s'introduisent à l'intérieur de l'ordinateur et donnent des renseignements à l'attaquant externe. Le code du cheval de Troie est généralement encapsulé dans un programme système nécessaire au fonctionnement de l'ordinateur.
- *Time bomb virus.* Ces virus sont liés à l'horloge du système et se déclenchent à une heure déterminée à l'avance.
- *Logical bombs*, ou bombes logiques. Ces virus se déclenchent lorsqu'un certain nombre de conditions logiques sont vérifiées.

Il est de plus en plus difficile de détecter les virus, les pirates essayant de les encapsuler dans des programmes innocents. Les parades à ces attaques sont nombreuses, quoique jamais complètement efficaces. La solution la plus simple consiste à se doter d'un anti-virus mis à jour régulièrement.

Conclusion

La sécurité dans Internet est un problème complexe pour la simple raison qu'elle n'a pas été introduite en même temps que les protocoles de base. Pour arriver à vendre des produits rapidement, les équipementiers ont laissé la sécurité de côté en pensant pouvoir facilement l'ajouter par la suite. En réalité, l'effort à faire pour ajouter les éléments de sécurité dans un environnement qui n'a pas été conçu pour cela pose de nombreux problèmes, dont les utilisateurs prennent conscience peu à peu.

Des efforts énormes sont déployés en ce sens depuis une dizaine d'années. Toutefois, même si l'on dispose maintenant de toute une batterie d'outils pour assurer la sécurité d'un réseau IP, ils ne sont généralement pas faciles à utiliser.

Références

Un excellent livre sur SSH :

D. J. BARRETT, R. SILVERMAN – *SSH, The Secure Shell*, O'Reilly, 2001

Un des très nombreux livres de U. Black, toujours fortement pédagogique :

U. D. BLACK – *Internet Security Protocols: Protecting IP Traffic*, Prentice Hall, 2000

Un bon livre pour apprendre à gérer la sécurité :

C. BRENTON – *Mastering Network Security*, Sybex, 1999

Un livre assez facile à lire et concret avec un grand nombre d'exemples sur les pare-feu et plus généralement sur la sécurité dans les réseaux IP :

W. R. CHESWICK, S. M. BELLOVIN, A. D. RUBIN – *Firewalls and Internet Security: Repelling the Wily Hacker*, Addison Wesley, 2003

Un excellent livre sur les réseaux privés virtuels de type IPsec :

C. DAVIS – *IPsec: Securing VPNs*, McGraw-Hill, 2001

IPsec sous toutes ses coutures :

N. DORASWAMY, D. HARKINS – *IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks*, Pearson Education, 2003

Petit livre très facile à lire mais introduisant parfaitement les éléments à connaître sur la sécurité dans Internet :

B. FABROT – *La Sécurité sur Internet*, Marabout, 2001

Très bon livre sur la problématique de la sécurité dans le commerce électronique :

W. FORD, M. S. BAUM – *Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption*, Prentice Hall, 2000

Très bonne introduction à IPsec et aux raisons de son succès :

S. FRANKEL – *Demystifying the IPsec Puzzle*, Artech House, 2001

Un excellent livre consacré à la sécurité sur Internet, qui aborde des aspects non classiques dans les stratégies à suivre pour sécuriser un réseau :

S. GHERNAOUTI-HÉLIE – *Sécurité Internet*, Dunod, 2000

Très bon livre pour entrer dans l'univers des virus :

D. HARLAY, R. SLADE – *Virus, définitions, mécanismes et antidotes*, Campus Press, 2002

Un livre très complet sur l'utilisation d'IPsec dans de nombreux contextes :

E. KAUFMAN, A. NEUMAN – *Implementing IPsec: Making Security Work on VPNs, Intranets, and Extranets*, Wiley, 1999

Un des nombreux livres de G. Held, qui porte sur la gestion des réseaux IP et plus particulièrement sur la sécurité :

G. HELD – *Managing TCP/IP Networks: Techniques, Tools and Security*, Wiley, 2000

Livre assez général qui aborde de nombreux éléments de la sécurité des réseaux IP :

O. KYAS – *Internet Security Risk Analysis, Strategies and firewall*, Thomson Computer Press, 1997

Livre à lire pour tous ceux qui s'intéressent à la sécurité sur Internet :

E. LARCHER – *L'Internet sécurisé*, Eyrolles, 2000

Un livre de base très complet, voire parfois un peu trop technique, pour entrer dans la sécurité par IPsec :

P. LOSHIN – *Big Book of IPsec RFCs: Internet Security Architecture*, Morgan Kaufmann Publishers, 1999

Un excellent livre sur les techniques de détection d'intrusion dans les réseaux :

S. NURTHCUTT – *Network Intrusion Detection, an analyst's handbook*, New Riders Publishing, 1999

Un livre consacré aux protocoles SSL et TLS :

E. RESCORLA – *SSL and TLS: Designing and Building Secure Systems*, Addison Wesley, 2000

Le Web pose des problèmes spécifiques en matière de sécurité. Le livre suivant leur est consacré :

A. D. RUBIN, D. GEER, M. J. RANUM – *Web Security Sourcebook*, Wiley, 1997

Les réseaux privés virtuels de type IPsec sont très utilisés pour gérer la sécurité. Le livre suivant en donne un panorama :

JA. M. TILLER, JI. S. TILLER – *A Technical Guide to IPsec Virtual Private Networks*, Auerbach Publications, 2000

Un excellent livre, qui présente les algorithmes de chiffrement de façon pédagogique :

W. STALLINGS – *Cryptography and Networks Security*, Prentice Hall, 1998

Très bon livre sur les pare-feu et leur utilisation dans le contexte des entreprises :

E. D. ZWICKY, *et al.* – *Building Internet Firewalls*, O'Reilly, 2000