Chapitre 10

La sécurité des réseaux

Au cours de ce chapitre, nous abordons principalement les différents aspects liés à la sécurité dans les réseaux et nous traitons le point particulier que représentent les usagers nomades. Nous commençons par les divers risques qu'encourent les entreprises et les menaces auxquelles elles sont exposées, en abordant exclusivement l'angle des réseaux. L'ISO a défini le vocabulaire des services et des mécanismes de sécurité : nous rappelons les définitions de l'authentification, l'intégrité, la non-répudiation, etc. Dans une seconde partie, nous donnons quelques exemples des solutions retenues actuellement pour faire face aux différents risques et menaces. À titre d'exemple, nous abordons le chiffrement, la signature numérique, les certificats, les réseaux privés virtuels, les pare-feux, etc. La panoplie des protections est très vaste, elle s'accroît avec la créativité des attaquants.

1 Gestion de la sécurité

Dans cette première section, nous définissons la sécurité des systèmes d'information et des réseaux, tout d'abord en termes de risques et de menaces qu'ils encourent. À partir de l'analyse de ces derniers, nous rappelons l'importance de la politique de sécurité. Nous introduisons ensuite les différents services de sécurité que l'ISO a normalisés et les mécanismes imaginés pour rendre ces services. Le chiffrement est l'un des mécanismes fondamentaux : il assure plusieurs services de sécurité différents. Nous illustrons les mécanismes les plus importants avec quelques exemples fréquemment utilisés aujourd'hui.

1.1 Risques, menaces et politique de sécurité

Risques et menaces sont deux concepts fondamentaux pour la compréhension des techniques utilisées dans le domaine de la sécurité. Le *risque* est une fonction de paramètres qu'on peut maîtriser à la différence de la *menace* qui est liée à des actions ou des opérations émanant de tiers. Dans un réseau, *a fortiori* dans un grand réseau, la sécurité concerne non seulement les éléments physiques (câbles, modems, routeurs, commutateurs…) mais aussi les éléments logiques, voire volatils, que représentent les données qui circulent. Le responsable de la sécurité doit analyser l'importance des risques encourus, les menaces potentielles et définir un plan général de protection qu'on appelle *politique de sécurité*.

Risques

Les risques se mesurent en fonction de deux critères principaux : la *vulnérabilité* et la *sensibilité*. La vulnérabilité désigne le degré d'exposition à des dangers. Un des points de vulnérabilité d'un réseau est un point facile à approcher. Un élément de ce réseau peut être très vulnérable tout en présentant un niveau de sensibilité très faible : le poste de travail de l'administrateur du réseau, par exemple, dans la mesure où celui-ci peut se connecter au système d'administration en tout point du réseau.

La sensibilité désigne le caractère stratégique d'un composant du réseau. Celui-ci peut être très sensible, vu son caractère stratégique mais quasi invulnérable, grâce à toutes les mesures de protection qui ont été prises pour le prémunir contre la plupart des risques. Exemples : le câble constituant le média d'un réseau local lorsqu'il passe dans des espaces de service protégés, l'armoire de sauvegarde des logiciels de tous les commutateurs du réseau…

On peut classer les risques en deux catégories : *structurels*, ils sont liés à l'organisation et la démarche d'une entreprise ; *accidentels*, ils sont indépendants de l'entreprise.

Enfin, selon les niveaux de sensibilité et de vulnérabilité, on distingue souvent quatre niveaux de risques, selon qu'ils sont *acceptables*, *courants*, *majeurs* ou *inacceptables*.

- *Acceptables.* Ils n'induisent aucune conséquence grave pour les entités utilisatrices du réseau. Ils sont facilement rattrapables : pannes électriques de quelques minutes, perte d'une liaison…
- *Courants.* Ce sont ceux qui ne portent pas un préjudice grave. Ils se traduisent, par exemple, par une congestion d'une partie du réseau. La mauvaise configuration d'un équipement peut causer la répétition des messages émis, un opérateur peut détruire involontairement un fichier de configuration…

- *Majeurs.* Ils sont liés à des facteurs rares. Ils causent des préjudices ou des dégâts importants, mais ils peuvent encore être corrigés. Un incendie a ravagé le centre de calcul d'une entreprise. La conséquence se traduit par le remplacement de l'ensemble du matériel, mais, heureusement, tous les logiciels et les données avaient été sauvegardés et archivés dans un local antifeu.
- *Inacceptables.* Ils sont, en général, fatals pour l'entreprise. Ils peuvent entraîner son dépôt de bilan. Exemple : la destruction du centre informatique et de l'ensemble des sauvegardes des programmes et données.

Menaces

On peut également classer les menaces en deux catégories selon qu'elles ne changent rien (menaces *passives*) ou qu'elles perturbent effectivement le réseau (menaces *actives*).

Les menaces passives consistent essentiellement à copier ou à écouter l'information sur le réseau, elles nuisent à la confidentialité des données. Dans ce cas, celui qui prélève une copie n'altère pas l'information elle-même. Il en résulte des difficultés à détecter ce type de malveillance, car elles ne modifient pas l'état du réseau. La méthode de prélèvement varie suivant le type de réseau. Sur les réseaux câblés, on peut imaginer un branchement en parallèle grâce à des appareils de type analyseurs de protocole ou une induction (rayonnement électromagnétique). Sur les faisceaux hertziens, des antennes captent les lobes secondaires des faisceaux ; dans les transmissions par satellites, des antennes avec systèmes de poursuite existent…

Les menaces actives nuisent à l'intégrité des données. Elles se traduisent par différents types d'attaques. On distingue le brouillage, le déguisement (modification des données au cours de leur transmission, modification de l'identité de l'émetteur ou du destinataire), l'interposition (création malveillante de messages en émission ou en réception).

Les niveaux de piratage sont très variables. La gamme des pirates s'étend de l'amateur sans connaissances particulières du réseau qu'il pénètre ou tente d'infiltrer au professionnel, souvent membre de l'entreprise et au courant des procédures du réseau. Les mécanismes de sécurité doivent donc prendre en considération aussi bien le sondage aléatoire, pratiqué par l'amateur à la recherche d'un mot de passe, que la lecture, aux conséquences désastreuses, du catalogue central des mots de passe, des codes de connexion ou des fichiers. Les menaces actives sont de nature à modifier l'état du réseau.

Les menaces dues aux *accidents* (statistiquement 26 % des causes) sont le fait d'incendies, d'inondations, de pannes d'équipements ou du réseau, de catastrophes naturelles… L'utilisation ou l'exploitation maladroite, la mauvaise conception ou la réalisation hasardeuse, le défaut de qualité… constituent les menaces dues aux *erreurs* (évaluées à 17 %). Les menaces dues à la *malveillance* (57 % dont 80 % sont d'origine interne) concernent les actes tels que le vol des équipements, les copies illicites de logiciels et de documents techniques, le sabotage matériel et l'attaque logique (virus, modification…), les intrusions et l'écoute, les actes de vengeance…

Politique de sécurité

La définition d'une politique de sécurité nécessite d'abord l'analyse des informations qui circulent ou qui sont stockées (analyse de leur importance pour l'entreprise, analyse du coût que représenterait leur perte) et celles des menaces qu'on peut objectivement envisager.

Les priorités de l'entreprise et sa stratégie influent sur le choix des procédures internes que devront respecter tous les utilisateurs. Il faut définir les mécanismes de protection à mettre en œuvre (les outils antivirus, les pare-feu, les *patches* ou programmes de correction des

systèmes et des applications utilisés) puis tous les outils de surveillance (depuis l'audit jusqu'au journal historique et la détection des intrusions). De nombreuses sociétés de service ont fait de la sécurité leur préoccupation première. Il est aujourd'hui évident pour une entreprise qu'il faut parler de la *sécurité du système d'information*, laquelle englobe le réseau.

Nous ne détaillerons ici que les aspects de la sécurité directement liés au réseau, sans aborder la protection contre le dégât des eaux, le contrôle d'accès physique aux bâtiments, la mise en place d'onduleurs et de générateurs pour maintenir l'alimentation électrique stable…

1.2 Services de sécurité

L'ISO a défini six services de sécurité : authentification, contrôle d'accès, confidentialité et intégrité des données, non-répudiation et protection contre l'analyse du trafic. Différents types de mécanismes (chiffrement, signature numérique, listes de contrôle d'accès, bourrage, notarisation…) servent pour assurer ces services. Ils diffèrent par leur sophistication, leurs coûts, les efforts nécessaires pour leur implantation, leur maintenance et leurs besoins en ressources humaines.

Authentification

Le service d'authentification garantit *l'identité* des correspondants ou des partenaires qui communiquent. On distingue deux cas d'authentification simple et un cas d'authentification mutuelle :

- *L'authentification de l'entité distante.* Elle garantit que le récepteur est celui souhaité. Son action peut intervenir à l'établissement de la communication ou pendant le transfert des données. Son objectif principal est la lutte contre le déguisement, également appelé *usurpation d'identité (spoofing)*.
- *L'authentification de l'origine.* Elle assure que l'émetteur est celui prétendu. Le service est inopérant contre la duplication d'entité. Comme le précédent, il s'agit d'authentification simple.
- *L'authentification mutuelle.* Elle assure que les deux entités émettrice et réceptrice se contrôlent l'une l'autre.

Le service d'authentification est inutilisable dans le cas d'un réseau fonctionnant en mode sans connexion : dans les réseaux, comme dans la vie courante, l'authentification nécessite un échange entre les deux partenaires.

Exemple

À la banque, pour prouver votre identité, vous montrez une carte nationale d'identité. Le guichetier effectue un rapide contrôle visuel, entre votre visage et la photo qui est sur la carte. Il y a bien échange entre vous et le guichetier. Un niveau de sécurité supplémentaire consiste à vous faire signer en présence du guichetier : celui-ci vérifie la signature manuscrite présente sur la carte. Dans les deux cas de cet exemple, le guichetier fait confiance aux autorités qui délivrent la carte d'identité pour avoir vérifié l'authenticité de votre identité. Si vous avez volé la carte d'identité, saurez-vous aussi ressembler à la photo et imiter la signature en temps réel ?

Contrôle d'accès

Le service de contrôle d'accès empêche l'utilisation non autorisée de ressources accessibles par le réseau. Par « utilisation », on entend les modes lecture, écriture, création ou suppression. Les ressources sont les systèmes d'exploitation, les fichiers, les bases de données, les applications… Pour contrôler les accès aux ressources, il faut d'abord authentifier les utilisateurs afin de s'assurer de leur identité qui est transportée dans les messages d'initia-

lisation et ensuite établir une liste des droits d'accès associés à chacun. L'annuaire LDAP fournit en général les données nécessaires à la mise en œuvre d'un tel mécanisme.

Confidentialité des données

Garantir la confidentialité des données empêche une entité tierce (non autorisée, le plus souvent en état de fraude passive) de récupérer ces données et de les exploiter. Seuls les utilisateurs autorisés doivent être en mesure de prendre connaissance du contenu des données. Un message ou un échange de messages a sa confidentialité garantie dès lors que tout utilisateur non autorisé qui aurait pu le récupérer ne peut pas l'exploiter. Il n'est pas obligatoire de mettre en place des procédures pour empêcher cette « récupération ».

Exemple

Certaines chaînes de télévision payantes sont transmises cryptées de telle sorte que seuls les possesseurs de décodeurs appropriés peuvent regarder leurs émissions favorites. Les autres peuvent toujours rester devant un écran zébré !

Intégrité des données

Garantir l'intégrité des données assure au récepteur que les données reçues sont celles qui ont été émises. Les données ont pu être altérées, de manière accidentelle ou de manière délibérée à la suite d'une fraude active. On distingue différents niveaux de service selon les mécanismes mis en œuvre. Peut-on détecter que des données ont été modifiées ? Si oui, peut-on récupérer les données initiales ? Sait-on détecter les données supplémentaires, insérées à tort ou délibérément ? Peut-on détecter les données manquantes et les récupérer ? Peut-on détecter que des données *a priori* correctes ne sont que des doublons de données déjà reçues ?

Par ailleurs, l'intégrité possède une portée plus ou moins grande (le message complet ou un champ spécifique du message seulement). Lorsque la communication a lieu en mode non connecté, seule la détection des modifications peut être mise en œuvre. Nous avons vu au chapitre 2 du livre les principes de la protection contre les erreurs : ajouter un bloc de contrôle d'erreur qui est le résultat d'un algorithme connu appliqué au message. Le récepteur refait le calcul sur le message qu'il a reçu et compare les deux blocs de contrôle d'erreurs. Il vérifie ainsi l'intégrité du message, cette seule méthode est insuffisante pour détecter des messages insérés dans un flux de données. Les protections mises en œuvre s'inspirent du même principe.

Non-répudiation

La non-répudiation de l'origine fournit au récepteur une preuve empêchant l'émetteur de contester l'envoi d'un message ou le contenu d'un message effectivement reçu. La non-répudiation de la remise fournit à l'émetteur une preuve empêchant le récepteur de contester la réception d'un message ou le contenu d'un message effectivement émis.

Exemple

Vous postez un courrier en « recommandé avec accusé de réception ». La Poste ajoute à votre courrier un document qui sera signé par le récepteur et qui sera ensuite renvoyé à l'expéditeur. Pour vous, la possession de cet accusé de réception interdit au récepteur de prétendre qu'il n'a rien reçu. La Poste joue un rôle d'intermédiaire entre vous et votre correspondant, elle rend le service de non-répudiation du courrier… Dans cette opération, elle ne vérifie pas votre identité et encore moins le contenu de votre lettre ! Votre correspondant peut soutenir avoir reçu une enveloppe vide.

Protection contre l'analyse de trafic

Le secret du flux lui-même empêche l'observation du flux de transmission de données, source de renseignements pour les pirates. Ce cas s'applique aux situations où on a besoin de garder la confidentialité sur l'existence même de la relation entre les correspondants.

1.3 Mécanismes de sécurité

Les exemples précédents viennent de la vie courante (banque, poste…). Dans la transmission de messages sur un réseau, il y a une énorme différence : un message électronique peut être dupliqué sans que rien ne permette la distinction entre l'original et celui qui est dupliqué puisque ce sont toujours des suites de données binaires. Il faut donc adapter les solutions de sécurité au monde électronique.

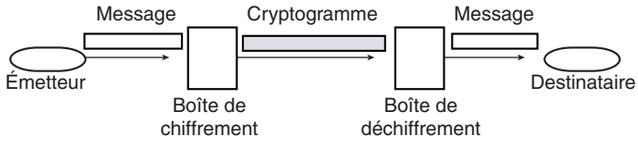
On a imaginé plusieurs mécanismes pour mettre en œuvre et offrir les services de sécurité énumérés précédemment. Il s'agit principalement du chiffrement – qui intervient dans presque tous les mécanismes –, de la signature numérique, des techniques d'utilisation d'identificateur et de mots de passe, de bourrage et de notarisation.

Chiffrement

Le chiffrement transforme tout ou partie d'un texte dit *clair* en *cryptogramme*, message chiffré ou protégé. Si une communication utilise des dispositifs de chiffrement, les données sont transmises sous une forme « brouillée », de manière qu'elles ne puissent être comprises par un tiers (voir figure 10.1).

Figure 10.1

Du message en clair au cryptogramme.



Le mécanisme de chiffrement émet un message X sous une forme secrète au moyen d'une clé K. L'émetteur dispose d'une fonction algorithmique E, qui, à X et K, associe $E(K, X)$. On utilise l'initiale E pour *Encryption.* Le récepteur reçoit $E(K, X)$ [message chiffré émis] et le déchiffre au moyen de sa clé K' avec sa fonction algorithmique de déchiffrement D, qui à $E(K, X)$ et K' associe X. On a alors :

$$D(K', E(K, X)) = X.$$

Les fonctions E et D peuvent être secrètes ou publiques. Il en est de même pour les clés K et K'. L'existence d'un déchiffrement tient à la définition de l'algorithme donnant E et D et à la méthode produisant et répartissant les clés K et K'. Les algorithmes utilisés ont évolué dans le temps, les premiers étaient symétriques, les plus récents sont asymétriques.

Chiffrement symétrique Historiquement, les premiers algorithmes de chiffrement étaient tels que $K = K'$ et $D = E^{-1}$. La clé K, unique, était secrète et l'algorithme du récepteur consistait à faire l'inverse de l'algorithme de l'émetteur : il suffisait de connaître la clé K. On parle alors de *chiffrement symétrique* car il n'y a qu'une clé. Les opérations de l'émetteur et celles du destinataire sont les inverses les unes des autres.

L'un des premiers systèmes connus est celui attribué à César. Il consistait simplement à substituer aux lettres du message d'autres lettres, qui se déduisent par simple décalage.

Exemple

Avec une clé de 15 (le décalage circulaire fait que le A est remplacé par P, B par Q, C par R… comme le montre le tableau 10.1).

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

Tableau 10.1
Code de César

Il devient simple de faire le chiffrement : VIVE LE MONDE DES RESEAUX devient KXKT AT BDCST STH GTHPJM. Un tel système ne résiste pas à la cryptanalyse c'est-à-dire au déchiffrement brutal sans la clé. La méthode utilisée étant une simple substitution, les fréquences d'apparition des lettres dans la langue utilisée restent respectées : ici il y a six T et au plus deux fois une autre lettre. Il est facile de penser que T code le E… et le reste suit.

Pour améliorer la robustesse du système, on a imaginé des méthodes qui « mélangent » les lettres et ne remplacent pas toujours une lettre par la même. De tout temps, on a assisté à une course-poursuite entre les concepteurs de systèmes de chiffrement et ceux qui cherchent à les casser avec des attaques brutes, des attaques statistiques, des analyses de plus en plus poussées.

Exemple

Imaginons une généralisation du code précédent avec une clé plus longue (3, 15, 21, 12, 5). La lettre qui remplace la lettre en clair est prise alternativement dans la première ligne du tableau 10.2 (clé 3), puis dans la deuxième (clé 15), puis dans la suivante (clé 21)… et ainsi de suite. Quand on arrive à la sixième lettre à coder, on reprend la clé 3.

Tableau 10.2
Code de César avec une clé plus longue

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E

VIVE LE MONDE DES RESEAUX devient YXQQ QH BJZIH SZE WHHZMZA. L'allongement de la clé limite les possibilités d'analyse statistique. Ici sur les quatre H, trois seulement représentent un E, de même que, sur les trois Z, deux seulement sont des E, quant aux trois Q successifs, ils codent V, E et L. En fait, pour casser le système, il faut trouver la longueur de la clé, car les probabilités d'apparition des lettres se retrouvent quand on hache le message en fonction de celle-ci.

Les algorithmes les plus courants du type symétrique sont DES (*Data Encryption Standard*) et ses variantes, RC4, 5 et 6, IDEA (*International Data Encryption Algorithm*) et AES (*Advanced Encryption Standard*). Nous verrons l'exemple du DES plus loin.

L'inconvénient d'un système de chiffrement symétrique aussi sophistiqué soit-il est que la clé *K* doit être transmise entre l'émetteur et le récepteur. Or si les correspondants ont choisi de faire du chiffrement, c'est en général parce qu'ils considèrent que le réseau n'est pas sûr : comment transmettre alors la clé ? On peut imaginer un transport physique de la clé par des moyens différents (valise diplomatique, par exemple). Quand deux correspondants ont déjà partagé une clé, ils l'utilisent pour transporter la nouvelle clé sur le réseau, et cette nouvelle clé est chiffrée. S'ils font confiance à un tiers et qu'ils ont des moyens de communication sûrs avec ce dernier, ils peuvent lui sous-traiter le problème de choix de la clé. Dans tous les cas, il faut définir un système de *distribution des clés*, dans la mesure où dans un réseau il n'y a pas deux correspondants seulement, mais des dizaines, des milliers

de communications différentes. Il faut une clé par communication où on veut assurer la confidentialité. Enfin, la non-répudiation est importante. Considérons un couple d'utilisateurs *A* et *B* qui se partageant une clé *K*. L'utilisateur *B* peut fabriquer des messages et faire croire que *A* les lui a envoyés !

Chiffrement asymétrique Une avancée considérable a été proposée par Diffie et Hellman[1] en 1976 avec le chiffrement *asymétrique* ou chiffrement à clés publiques. Dans ce cas, la clé se compose de deux éléments, l'un est public (publié dans un annuaire, par exemple), l'autre est secret (et jamais transmis). Pour envoyer un message confidentiel à un utilisateur *A*, on lui demande l'élément public de sa clé (ou on le lit dans l'annuaire). On chiffre le message avec celui-ci. L'utilisateur *A* est capable de déchiffrer le message en utilisant l'élément secret de sa clé. Lui seul est capable de le faire puisqu'il est le seul à connaître cet élément. En effet, la connaissance de l'élément public de la clé ne permet pas de retrouver l'élément secret.

Le chiffrement asymétrique a une propriété remarquable : un message chiffré avec l'élément public de la clé se déchiffre avec l'élément secret de la même clé et, à l'inverse, un message chiffré avec l'élément secret de la clé se déchiffre avec l'élément public de la même clé. Cette propriété sert pour l'authentification et la confidentialité. En effet, dans le scénario précédent, l'utilisateur *A* reçoit un message confidentiel que n'importe qui a pu lui envoyer. Pour s'authentifier, l'émetteur envoie son identité, le message et un bloc de données supplémentaire calculé avec un algorithme connu à partir du message lui-même et chiffré avec l'élément secret de sa propre clé. Le tout est chiffré avec la clé publique de *A*. Quand *A* déchiffre le message avec sa clé secrète, il découvre l'identité de l'émetteur, le message et un bloc de données qu'il peut maintenant déchiffrer en demandant à l'émetteur la partie publique de sa clé (ou en la lisant dans l'annuaire). Une fois le bloc de données déchiffré, *A* peut vérifier que celui-ci est cohérent avec le message.

Les algorithmes les plus connus du type asymétrique sont RSA (*Rivest Shamir Adleman*) et ECC (*Elliptic Curve Cryptosystem*). Ils utilisent des éléments de mathématiques de très haut niveau.

L'intérêt principal du chiffrement asymétrique est qu'il n'y a pas de clé à transmettre. Par contre, les calculs à effectuer pour chiffrer et déchiffrer sont plutôt longs. Cela peut être dissuasif pour des applications à fortes contraintes temporelles ou pour des équipements (comme les capteurs) qui ont de très faibles capacités de calcul et de stockage. La tendance actuelle est d'utiliser ces algorithmes dans la partie de contrôle de la communication. Ils servent à distribuer les clés qui seront ensuite utilisées pour chiffrer les données avec des algorithmes classiques symétriques.

Place du chiffrement Le mécanisme de chiffrement existe à trois niveaux : *voie par voie*, *réseau*, *de bout en bout*. L'ensemble repose, dans tous les cas, sur un algorithme donné, une clé ou un couple de clés associées et un mécanisme de distribution des clés.

Le chiffrement *voie par voie* est le résultat de la mise en place de boîtes noires sur les supports de transmission, qui laissent les données en clair au niveau des hôtes et des nœuds du réseau. Le message est chiffré/déchiffré indépendamment à chaque changement de support de transmission. Le chiffrement est alors mis en œuvre dans la couche Liaison de données. Un tel chiffrement est transparent à l'utilisateur qui n'a pas connaissance des procédures internes de l'opérateur du réseau.

1. Bailey Whitfield Diffie (né en 1944) et Martin Hellman (né en 1945) sont des chercheurs américains, pionniers de la cryptographie à clé publique.

Dans le chiffrement au niveau *réseau*, le service est fourni, entre deux sites donnés, par deux équipements spécialisés, placés sur chacun des sites. Ces équipements sont situés au niveau des routeurs de sortie : tout le flux circulant entre les deux sites est chiffré et traverse le réseau utilisé avec confidentialité. L'inconvénient principal est que toutes les données sont chiffrées alors quelles ne le nécessitent pas forcément.

Le chiffrement *de bout en bout* laisse en clair les informations de routage, seules les données constituant l'information transmise sont chiffrées. Il appartient logiquement à la couche Présentation telle qu'elle a été conçue par l'ISO et il est mis en œuvre dans les applications du modèle TCP/IP.

Services rendus par l'usage du chiffrement Il est aisé de comprendre que le chiffrement assure la confidentialité des échanges. Dans le cas des algorithmes à clés publiques, nous avons vu qu'il participe également à l'authentification : seul celui qui possède l'élément secret de la clé peut faire un certain nombre d'opérations. Dans l'exemple que nous avons cité, l'utilisateur *A* qui reçoit un message confidentiel peut le déchiffrer mais il n'a aucune information quant à celui qui l'a émis. Un mécanisme supplémentaire de signature est utilisé pour garantir l'identité de celui qui émet.

Signature numérique

La signature numérique consiste à utiliser un chiffrement particulier appelé chiffrement *irréversible*. Celui-ci transforme un message (*a priori* long) en un bloc de données (de petite taille) tel qu'il est impossible de reconstruire le message à partir du bloc. Les algorithmes utilisés sont appelés *fonction de hachage* ou *fonction de condensation*. Le bloc est appelé *condensé* ou *signature*. Une bonne fonction de hachage doit produire des condensés différents pour des messages différents : si deux messages différents avaient le même condensé, il serait possible pour un utilisateur malveillant de substituer un message à l'autre, tout en conservant le condensé correct. Cela rend la modification du message indétectable.

On obtient une signature numérique en appliquant (avec une clé) la fonction de hachage au message transmis. Celui-ci devient *signé*. On envoie le message et sa signature. Le propre de la signature est qu'elle est vérifiable par ceux qui possèdent la clé, mais inimitable.

Les algorithmes les plus connus du type irréversible sont MD5 (*Message Digest5*) et SHA1 (*Secure Hash Algorithm1*). Nous verrons leurs principes plus loin.

On garantit l'intégrité d'une unité de données ou d'un champ spécifique d'une unité de données par les codes de contrôle cryptographique, dont le mécanisme est identique à celui des signatures numériques. L'intégrité d'un flot de données peut être assurée par le même mécanisme de cryptographie auquel s'ajoutent des codes de détection d'erreurs ainsi que la numérotation des unités de données par horodatage.

Mots de passe

Lorsque les entités homologues et les moyens de communication sont sûrs, l'identification des entités homologues peut se faire par un identificateur d'utilisateur (*login*) et un *mot de passe*. La sécurité ne peut pas se fonder sur l'identificateur seul. Celui-ci est habituellement de notoriété publique, tel le numéro d'identification de l'employé. De plus, on ne peut pas le changer facilement du fait que beaucoup d'informations s'y rattachent.

Dans certaines applications, l'utilisateur ne connaît même pas son mot de passe qui est inscrit dans une carte magnétique contenant un NIP (numéro d'identification personnel). Dans d'autres applications, seul l'utilisateur connaît son numéro, et une fonction lui permet de changer son mot de passe. Le cas des guichets bancaires est particulier : le client doit introduire une carte contenant son code, plus une clé secrète.

Le responsable de la sécurité doit porter une attention particulière au protocole qui transporte le mot de passe et au fichier système qui stocke les mots de passe des utilisateurs : inutile de mettre en place un système d'identification avec identificateur et mot de passe si ceux-ci circulent en clair dans le réseau. Lorsque les moyens de communication ne sont pas sûrs, les mots de passe ne suffisent plus à réaliser le mécanisme ; il faut alors y adjoindre des procédures de chiffrement.

Liste de contrôle d'accès

Le mécanisme des listes de contrôle d'accès (ACL, *Access Control List*) utilise l'identité authentifiée des entités et des informations fiables pour déterminer leurs droits d'accès au réseau ou aux ressources sur le réseau. De plus, il est susceptible d'enregistrer sous forme de trace d'audit et de répertorier les tentatives d'accès non autorisées. Tout utilisateur qui se trompe dans son mot de passe laisse une trace. Il est ainsi possible de détecter les programmes automatiques qui cherchent à pénétrer le système en essayant tous les mots de passe. Les informations utilisées sont : les listes de droits d'accès, maintenues par des centres, les mots de passe, les jetons de droits d'accès, les différents *certificats* (voir plus loin), les libellés de sensibilité des données.

Le mécanisme de contrôle d'accès peut avoir lieu aux deux extrémités de la communication (équipement d'accès et ressource du réseau).

Bourrage et contrôle de routage par gestion dynamique de la bande passante

Le bourrage simule des communications dans le but de masquer les périodes de silence et de banaliser les périodes de communication réelles. Cela évite d'attirer l'attention des pirates lors des démarrages de transmission.

On obtient un mécanisme de bourrage en envoyant, entre deux émissions de messages utiles, des séquences de messages contenant des données dépourvues de sens. De plus, pour mieux créer l'illusion des vrais messages, le générateur de messages respecte la fréquence des lettres et des digrammes[2] de l'alphabet employé.

Enfin, après détection d'une attaque sur une route donnée, ou tout simplement pour prévenir cette attaque, les systèmes d'extrémités ou les réseaux peuvent, par le mécanisme de gestion dynamique de la bande passante, sélectionner une route plus sûre. Dans certains cas, la modification périodique est programmée afin de déjouer toutes les tentatives malveillantes.

Remarque

Les opérateurs de téléphonie mobile utilisent un tel mécanisme : l'allocation de la sous-bande de fréquences, intervalle de temps après intervalle de temps, est dynamique. Une communication n'occupe pas toujours la même sous-bande de fréquences. Cela fournit deux avantages. Si une sous-bande de fréquences est moins bonne que les autres, elle se retrouve affectée de temps en temps aux différentes communications, qui ne souffrent ainsi pas trop de sa piètre qualité. On optimise la qualité de transmission. Ensuite, il est difficile d'écouter une communication puisqu'il faut connaître l'algorithme d'affectation des sous-bandes à la communication.

2.Un digramme est un ensemble de deux lettres.

Notarisation

La *notarisation* apporte une garantie supplémentaire : les entités font confiance à un tiers qui assure l'intégrité et atteste de l'origine, la date et la destination des données. Le processus sous-entend que ce tiers doit acquérir les informations par des voies de communication très protégées. L'authentification, par exemple, peut être sous-traitée à un tiers de confiance : un utilisateur lui fournit une preuve de son identité et obtient de sa part un certificat numérique (le tiers devient alors une *autorité de certification*). Pour que le système soit complètement sûr, il faut contrôler les autorités de certification elles-mêmes : dans l'Union européenne, les gouvernements assurent ce contrôle.

2 Aspects opérationnels de la sécurité

Les protocoles de la pile TCP/IP n'ont pas été conçus avec des objectifs de sécurité. En particulier, dans leur conception initiale, il n'y avait aucun chiffrement, pas de contrôle d'intégrité des données (protocoles IP et UDP), pas d'authentification des extrémités… Avec Telnet, l'usager se croit protégé par un identifiant et un mot de passe : ces derniers circulent en clair dans le réseau. Le protocole de gestion de réseaux SNMP ne prévoyait, dans sa version 1, aucun mécanisme d'authentification. Or, une commande *get* permet d'obtenir la copie complète (*dump*) de la configuration d'un routeur, par exemple. Quant à la commande *set*, elle peut effacer cette même configuration. Pour éviter des configurations sauvages malveillantes, on a alors imaginé d'invalider la commande *set* sur les différents routeurs du marché, privant de ce fait l'administrateur autorisé de ses possibilités d'actions à distance. Le protocole ICMP initialement conçu comme outil de contrôle d'IP pouvant lui aussi renseigner un pirate, il devenait prudent d'interdire les messages de ce protocole !

On a donc plaqué de nombreuses solutions de sécurité sur la pile TCP/IP, à différents niveaux, du fait de cette absence de conception initiale. On peut les considérer par bien des aspects comme du bricolage. Dans cette section, nous analysons quelques attaques bien connues et décrivons les principales parades utilisées.

2.1 Quelques attaques bien connues

Les utilisateurs aux intentions malveillantes ont déployé de nombreuses attaques sur des sites commerciaux ou des sites de grandes sociétés et organismes. Les principales sont le déni de service (DoS, *Denial of Service*), l'inondation de requêtes d'ouverture TCP (*SYN flooding*), la dissimulation ou l'usurpation d'adresses IP (*IP Spoofing*).

Le déni de service (DoS, *Denial of Service*)

Une attaque en déni de service consiste à bloquer une machine cible en lui envoyant des requêtes inutiles. Cela l'empêche de rendre le service pour lequel on l'a installée. L'attaque la plus simple est l'inondation par des *ping* (messages *ICMP Echo Request*) ou des messages ICMP avec beaucoup de données forçant les différents intermédiaires à traiter la fragmentation. La machine cible passe son temps à répondre aux sollicitations reçues et n'a plus de disponibilité pour son propre service.

L'inondation de requêtes d'ouverture (*SYN Flooding*)

Une demande d'ouverture de connexion TCP (segment avec drapeau SYN mis à 1) provoque une réponse avec les drapeaux SYN et ACK mis à 1 puis une attente du troisième seg-

ment avec seulement le drapeau ACK mis à 1. L'attaque par inondation de requêtes d'ouverture consiste à envoyer à une machine cible un grand nombre de segments avec drapeaux SYN mais sans jamais transmettre le troisième segment. La machine cible réserve vainement des ressources à chaque requête d'ouverture et passe son temps à gérer les temporisateurs d'attente du troisième segment qui confirme l'ouverture.

Dissimulation d'adresse IP (*IP Spoofing*)

Le datagramme IP transporte l'adresse IP de l'émetteur et, en l'absence d'un mécanisme d'authentification de l'adresse, il est impossible de vérifier qui a émis avec cette adresse. Un pirate veut attaquer un réseau dont il connaît l'adresse IP : il usurpe l'une de ces adresses et l'utilise comme adresse source. Il y a toutes les chances pour que son datagramme soit considéré comme un datagramme normal du réseau… sauf s'il se présente, venant d'Internet, à la porte d'entrée du réseau et que l'administrateur a prévu qu'un message avec une adresse IP d'émetteur interne ne puisse pas provenir de l'extérieur.

Autres attaques

Les pirates ont toujours beaucoup d'imagination : utiliser un port (ouvert) proposé pour un protocole donné avec un autre protocole ce qui donne des possibilités de manipulations sur la machine cible ; voler des sessions (*hijacking*) TCP ouvertes de l'intérieur, profiter des failles de sécurité sur une machine pour l'utiliser ensuite comme source et profiter des droits d'accès de celle-ci (rebond). Le rejeu est également une attaque possible, il consiste à réinjecter dans le réseau des messages corrects (chiffrés, signés…) qui ont déjà été transmis.
Nous pourrions ranger dans cette catégorie les chevaux de Troie ou les vers…

2.2 Les principales parades

Les techniques de sécurité décrites ci-dessus réduisent les risques de manière significative. Elles ne les éliminent pas totalement. Des mécanismes de détection d'intrusion ou de violation doivent être implantés pour surveiller de façon continue le réseau. Le flot général des messages, des événements et des alarmes, est analogue à celui de la gestion des pannes. Cependant, des actions spécifiques sont à prendre pour la gestion de la sécurité. Elles doivent, en particulier, avoir un impact minimal sur le fonctionnement opérationnel du réseau et maximiser les chances de démasquer le pirate.

Les journaux d'historiques (*logs*) sont les sources d'information les plus utiles. Ils contiennent tous les événements et incidents de communication présélectionnés par le responsable de la sécurité (refus d'accès, tentatives de connexion avec échec…), les identificateurs des usagers, émetteur, récepteur avec une indication de l'initiateur de la connexion, la date, l'heure… les ressources impliquées dans la communication, les mots de passe et/ou clés utilisées, les fonctions de sécurité appelées, manuellement ou automatiquement. Les équipements de collectes de mesures et les instruments de gestion des pannes participent à la gestion de la sécurité. Certains équipements sont spécifiques : les boîtes de chiffrement, les contrôleurs d'accès, les contrôleurs d'authentification…

Parmi les outils classiques que nous détaillons dans cette section, nous pouvons citer le standard DES, l'algorithme RSA, l'algorithme MD5, Kerberos et les pare-feu (*firewalls*). La mise en œuvre de réseaux privés virtuels (VPN, *Virtual Private Network*) fait l'objet d'une section séparée.

Chiffrement DES (*Data Encryption Standard*)

Le mécanisme de chiffrement le plus utilisé est fondé sur le standard américain DES (*Data Encryption Standard*) adopté par le NIST (*National Institute for Standards and Technologies*) en 1977. L'algorithme DES découpe les messages de données en blocs de 64 bits. Il transforme chaque bloc en un autre bloc de 64 bits à l'aide d'une clé (limitée par les instances fédérales américaines à 56 bits). Le même algorithme et la même clé servent pour le déchiffrement. Il s'agit d'un algorithme dont toutes les opérations sont connues (permutations, additions, substitutions). Sa sécurité réside dans la clé secrète, c'est-à-dire dans la complexité des calculs nécessaires pour analyser toutes les clés possibles, en l'absence de toute autre information. Le principal problème est que les progrès technologiques de l'informatique rendent aujourd'hui possibles des calculs qui étaient infaisables, il y a quelques années seulement. Pour augmenter la sécurité d'un tel système, on utilise fréquemment plusieurs opérations en cascade (double DES, voire triple DES), avec des clés différentes.

Principe du DES Les grandes lignes de l'algorithme sont les suivantes. La première étape fractionne le texte en blocs de 64 bits (ou 8 octets). On effectue, sur chaque bloc, une permutation initiale. Celle-ci est telle que le 58^{e} bit d'un bloc devient le premier, le 50^{e} devient le deuxième, le 42^{e} le troisième, etc., conformément au vecteur suivant (58, 50, 42, 34, 26, 18, 10, 2, 60, 52, 44, 36, 28, 20, 12, 4, 62, 54, 46, 38, 30, 22, 14, 6, 64, 56, 48, 40, 32, 24, 16, 8 ; 57, 49, 41, 33, 25, 17, 9, 1, 59, 51, 43, 35, 27, 19, 11, 3, 61, 53, 45, 37, 29, 21, 13, 5, 63, 55, 47, 39, 31, 23, 15, 7). On découpe alors chaque bloc en deux parties, gauche et droite, nommées *G* et *D.* On exécute ensuite seize étapes successives de permutation et de substitution sur les parties *G* et *D*. Soit G_0 et D_0, l'état initial de ces deux parties. Dans chacune des seize étapes suivantes, les blocs G_i et D_i (i = 0 à 15) subissent des opérations d'expansion (ils sont transformés en blocs de 48 bits par recopie de 16 de leurs bits disséminés dans les 32 de départ selon un ordre connu) puis de OU exclusif avec la clé K_i et enfin de substitution qui reconstituent un bloc de 32 bits. On reconstitue ensuite un bloc de 64 bits à partir de ses deux parties droite et gauche. On exécute enfin une permutation finale qui est l'inverse de la permutation initiale.

Algorithme RSA (*Rivest, Shamir, Adleman*)

L'algorithme RSA (*Rivest*, *Shamir*, *Adleman*, du nom de ses concepteurs) est un algorithme à clé publique conçu au MIT en 1978. Il utilise des problèmes NP complets : par exemple, la décomposition d'un nombre en facteurs premiers (le nombre en question possède cent ou deux cents chiffres…). Celui qui cherche à protéger ses communications rend ce nombre public (dans un annuaire sur une page Web…), mais le résultat de la décomposition est connu de lui seul. Même si un espion intercepte un message, il ne peut donc pas le déchiffrer. Un tel algorithme permet au seul récepteur autorisé de lire les messages qui lui sont destinés.

Principe de l'algorithme RSA Cet algorithme est fondé sur la théorie des nombres. L'utilisateur choisit deux nombres premiers, p et q, chacun plus grand que 10^{100}, et calcule $n = p^{*}q$ et $\phi\ (n) = (p-1)^{*}(q-1)$. Il choisit un nombre d premier avec $\phi\ (n)$ et cherche un nombre e tel que $e^{*}d = 1\ (\text{mod}\ \phi\ (n))$. La clé publique est le couple (e, n), la clé secrète le couple (d, n).

Considérons un message à chiffrer qui doit être expédié à un utilisateur qui a publié sa clé (e, n). Le message en clair est découpé en une suite de blocs de telle sorte que chaque bloc en clair M soit un nombre inférieur à n. Il suffit de prendre des blocs de k bits, où k est le plus grand nombre entier tel que 2^{k} est inférieur à n.

Pour chiffrer le bloc M, on calcule $C = M^e \pmod n$ en utilisant les deux éléments publics de la clé (e, n) et on envoie C.

L'utilisateur cherche à déchiffrer le message C reçu. Il calcule $C^d \pmod n$ utilisant les deux éléments secrets de sa clé (d, n). On admettra le résultat : $C^d \pmod n = M$. Lui seul peut faire ce calcul : cela garantit la confidentialité du message transféré.

Exemple

Soit $p = 11$ et $q = 17$ d'où $n = 187$ et $\phi(n) = 160$. Choisissons $d = 7$, cette valeur convient puisque 7 et 160 n'ont pas de facteurs communs. L'équation $e^*7 = 1 \pmod{160}$ donne $e = 23$ puisque $23^*7 = 161 = 160 + 1$.

Pour chiffrer le message $M = 88$, l'émetteur calcule $88^7 \pmod{187}$ soit 11 et envoie ce « message ». Le récepteur qui connaît sa clé secrète calcule $11^{23} \pmod{160}$ et il trouve 88.

Signatures MD5 (*Message Digest 5*) et SHA1 (*Secure Hash Algorithm 1*)

Également conçue par Rivest, la signature numérique utilise un algorithme à clé publique pour la confidentialité et l'authentification. MD5 (*Message Digest 5*, défini en 1992 dans la RFC 1321, successeur de MD4…) prend en entrée des messages de longueur quelconque. Il découpe ces messages en blocs de 512 bits et produit en sortie un résultat de 128 bits grâce à une fonction de condensation. Les calculs utilisent des opérations simples, faites sur des blocs de 32 bits, pour une implémentation rapide. On a malheureusement découvert des failles de sécurité dans la fonction de hachage utilisée par MD5 : deux documents totalement différents peuvent avoir la même signature. SHA1 (*Secure Hash Algorithm 1*), un algorithme plus récent, fut alors préféré à MD5. SHA1 utilise une fonction de hachage qui produit un résultat de 160 bits. Cet algorithme possède les mêmes faiblesses que le précédent ! La tendance aujourd'hui est aux signatures de plus en plus longues : SHA256 produit une signature sur 256 bits, SHA384 (resp. SHA512) des signatures de 384 (resp. 512) bits.

PGP (*Pretty Good Privacy*)

Le système PGP (*Pretty Good Privacy*) sert pour la protection du courrier électronique. Conçu par P. Zimmerman en 1991, il avait pour objectif de proposer une solution simple pour assurer la sécurité de la messagerie électronique de personne à personne. Après bien des déboires (liés à la législation américaine sur l'exportation et à la protection industrielle de certains algorithmes qu'il utilisait), il est maintenant largement répandu grâce à sa version libre téléchargeable. Chacun des utilisateurs génère sa propre paire de clés (publique/privée) et distribue sa clé publique à ses interlocuteurs de confiance. Il est possible de crypter les messages (avec la clé publique du récepteur) et de les signer (avec la clé secrète de l'émetteur).

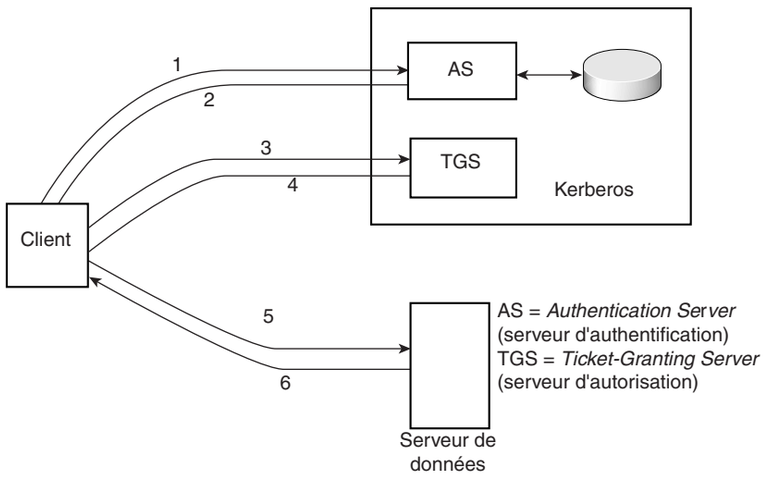
Kerberos

Kerberos est un service d'authentification développé au MIT pour fournir des services de sécurité dans un environnement client/serveur distribué. Le principe de fonctionnement est illustré à la figure 10.2. Pour utiliser un service, un client doit tout d'abord fournir auprès du serveur d'authentification un *certificat*, preuve de son identité et de ses droits. Il reçoit en retour des données qui ont une durée de vie limitée : un *ticket* et une *clé*. Armé de ces données, il adresse au serveur d'autorisation un message chiffré et daté contenant une demande d'autorisation d'accès à un serveur donné et reçoit en retour un *nouveau ticket* et une *nouvelle clé*. Il utilise ces dernières informations pour se connecter sur le serveur de

données, qui vérifie la validité des informations fournies. L'algorithme de chiffrement utilisé est le DES.

Figure 10.2

Serveur de sécurité Kerberos.



Kerberos est hébergé dans une station du réseau. Il est responsable de la génération de toutes les clés, gère les certificats d'identité et les tickets d'autorisation. Tous les serveurs de données et les applications du réseau doivent être déclarés auprès de Kerberos. Celui-ci limite le nombre d'authentifications mais ne gère pas l'accès aux fichiers ordinaires. Il suppose que l'environnement est un réseau local non sécurisé utilisant les protocoles de la famille TCP/IP.

Notons que la durée de vie limitée du ticket (souvent, quelques heures seulement) minimise les problèmes liés au vol. De plus, les tickets contiennent l'adresse IP du client, ils ne peuvent donc pas être utilisés sur une autre machine… à moins que le pirate ne fasse en plus de l'usurpation d'adresse IP. Une attaque sur le serveur authentification est difficile car Kerberos ajoute un identifiant à toute requête pour éviter les attaques par rejeu. Enfin, les serveurs conservent l'historique des communications et détectent facilement un envoi falsifié.

Certificats numériques *X.509*

La norme de l'ISO *X.509*[3] régit les certificats numériques. Ceux-ci contiennent le nom de la personne (ou de l'institution) pour lequel le certificat a été émis, la clé publique de la personne, la référence à l'algorithme utilisé par la personne pour sa signature numérique, la durée de validité du certificat. Chaque certificat, signé par l'autorité avec sa clé privée, possède un numéro de série unique géré par l'autorité de certification elle-même. Le certificat numérique est donc une façon d'obtenir (avec sécurité) la clé publique dont on a besoin pour vérifier une signature.

Une autorité de certification peut révoquer un certificat avant sa date d'expiration, il est donc recommandé de vérifier la liste des révocations pour être sûr que le certificat est valable.

Des entreprises comme Verisign proposent la délivrance des certificats numériques avec authentification, protection d'identité. Dans certains cas, les entreprises se contentent d'émettre elles-mêmes leurs propres certificats !

3. Remarquons qu'il s'agit d'un exemple rare de norme ISO qui a survécu au raz de marée TCP/IP.

Pare-feu ou *firewall* et zone démilitarisée

À ses origines, Internet fut conçu par et pour une communauté de chercheurs qui s'échangeaient librement des données, en absence de toute considération mercantile. Aujourd'hui, il occupe une telle place dans la vie professionnelle et dans la vie privée que pas un jour ne se passe sans que de nouveaux virus ou d'autres programmes malveillants apparaissent pour nuire à un maximum de machines ou de réseaux. Chaque ordinateur et, *a fortiori*, chaque réseau doivent se protéger contre ces attaques incessantes ; les entreprises ou les particuliers connectés en permanence doivent donc être particulièrement vigilants. Le pare-feu est un équipement matériel ou un logiciel qui surveille les paquets échangés avec le réseau ; c'est l'une des protections indispensables à installer sur une machine ou dans un réseau. La zone démilitarisée ou DMZ (*DeMilitarized Zone*) fait office de tampon ou de sas entre le réseau à protéger et le monde extérieur.

Le pare-feu Un pare-feu préserve le réseau des attaques en filtrant les paquets qui y circulent. Ce filtrage doit offrir en toute transparence aux utilisateurs du réseau d'entreprise tous les services dont ils ont besoin à l'extérieur. Il doit protéger les accès aux applications et aux données à l'intérieur du réseau d'entreprise. Le pare-feu fonctionne principalement grâce à un ensemble de règles. Celles-ci définissent les connexions autorisées (*allow*) et celles qui sont interdites (*deny*). Dans certains cas, le pare-feu peut rejeter une demande extérieure, sans même prévenir l'utilisateur concerné (*drop*). Les règles de refus peuvent être implicites (on interdit les communications qui n'ont pas été expressément autorisées)

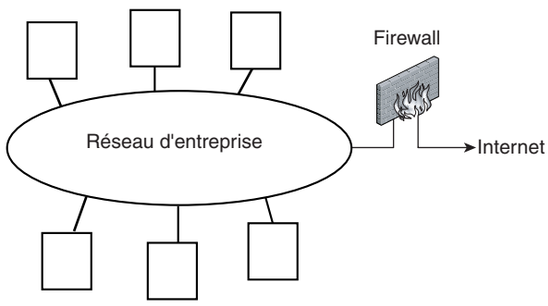
ou explicites (on n'interdit que ce qui a été spécifiquement interdit). Si la première méthode est la plus sûre, elle oblige à une définition exhaustive et précise des interdictions ; la seconde peut entraîner des failles de sécurité.

Avant de laisser un paquet entrer dans le réseau, le pare-feu en analyse l'en-tête. En particulier, il regarde les adresses IP source et destination. Il doit aussi examiner le contenu du paquet, notamment l'en-tête du segment transporté : protocole concerné (TCP ou UDP) et numéro du port (pour savoir quel type de service est demandé). La plupart des pare-feu sont configurés de manière à filtrer les communications selon le port utilisé. Cette méthode, appelée *filtrage dynamique*, ne suffit pas pour se protéger efficacement contre les failles de sécurité des applications elles-mêmes. Il faut ajouter un filtrage au niveau applicatif – un *proxy* – qui s'interpose entre le réseau et les applications.

Le pare-feu à séparation de réseaux (*dual homed firewall*) est illustré à la figure 10.3. C'est un routeur qui possède deux cartes réseaux et sépare physiquement le réseau d'entreprise d'Internet : tout le trafic interréseau passe par le pare-feu qui peut exécuter son filtrage sur chaque requête entrante ou sortante…

Figure 10.3

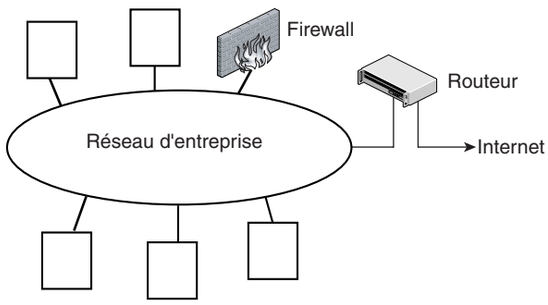
Un pare-feu qui sépare physiquement le réseau d'entreprise du réseau Internet.



Le pare-feu peut être une machine du réseau, distincte du routeur qui assure l'accès à Internet. On parle alors de *screened host firewall*, de pare-feu au fil de l'eau ou encore de bastion (voir figure 10.4). C'est le routeur qui agit activement en faisant transiter tout le trafic venant d'Internet vers la machine pare-feu. Inversement, il bloque tout trafic destiné à Internet qui est émis par une machine quelconque du réseau autre que le pare-feu. Les machines internes du réseau doivent donc connaître le pare-feu et lui adresser tout leur trafic effectivement destiné à Internet.

Figure 10.4

Un pare-feu au fil de l'eau vers lequel est détourné tout le trafic concernant Internet.

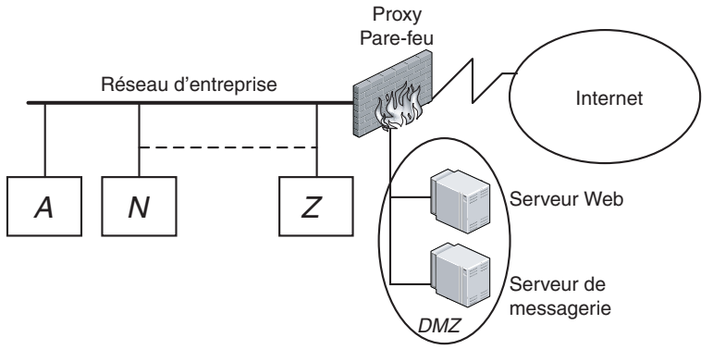


Un pare-feu n'est pas un dispositif d'une sécurité absolue. Sa protection est efficace s'il est bien configuré et si toutes les communications avec l'extérieur passent par lui (les postes nomades utilisant un modem pour accéder directement à Internet sont des failles de sécurité potentielles). Enfin, il faut surveiller très attentivement l'historique des connexions pour détecter toute tentative d'intrusion et modifier le paramétrage du pare-feu dès que de nouveaux modes d'attaque ont été signalés dans les bulletins de sécurité émis par différents organismes comme le CERT (*Computer Emergency Response Team*).

La zone démilitarisée (DMZ, *DeMilitarized Zone*) Les réseaux d'entreprise sont rarement monolithiques et comptent plusieurs parties, isolées les unes des autres par des pare-feu. On utilise une DMZ pour rendre accessible depuis l'extérieur un ensemble de services : serveur de messagerie, serveur FTP, portail Web… Le cloisonnement qui découle de ce choix implique une modification de l'architecture du réseau évoquée à la section précédente. La figure 10.5 montre où se situent la DMZ et le pare-feu par rapport au réseau de l'entreprise. Il faut définir, pour chaque zone du réseau, quels sont les flux autorisés et les flux interdits avec le monde extérieur.

Figure 10.5

Cloisonnement d'un réseau d'entreprise et utilisation du pare-feu.



2.3 RÉSEAUX PRIVÉS VIRTUELS

On désigne par *réseau privé virtuel* (VPN, *Virtual Private Network*) un réseau d'entreprise sécurisé, constitué de plusieurs sites reliés par Internet. La traversée d'Internet est vue comme un *tunnel*, dans lequel les données de l'entreprise sont chiffrées et transitent d'un bout à l'autre. L'entreprise ne peut avoir connaissance des autres données qui circulent sur les liaisons empruntées. Pour mettre en œuvre ce mécanisme de tunnel, on utilise un protocole spécial pouvant assurer plusieurs services selon les besoins de l'entreprise : confidentialité, intégrité des données, authentification des machines d'extrémité. Le principal protocole de tunnel est utilisé au niveau réseau : il s'agit d'IPSec (*IP Security*), une version sécurisée d'IP définie par la RFC 2246. L'entreprise reçoit donc le même service que si les liaisons lui appartenaient. C'est pourquoi on parle de *réseau virtuel*. Une autre solution consiste à utiliser des protocoles d'authentification et de chiffrement tels que SSL (*Secure Socket Layer*) qui protègent les échanges de données selon les besoins des utilisateurs et de leurs applications. Rebaptisée TLS (*Transport Layer Security*), cette solution est modulaire et situe la sécurité au niveau de la couche Transport. Nous décrivons ci-après les deux solutions IPSec et TLS puis nous proposons une comparaison.

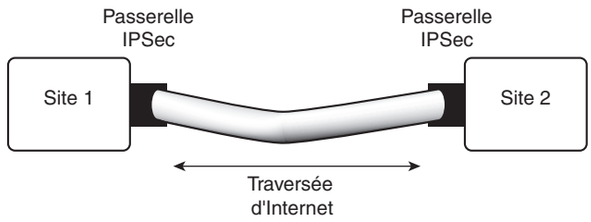
IPSec (*IP Security*)

On a conçu IPSec (*Internet Protocol Security*) pour sécuriser le protocole IPv6. La lenteur de déploiement de ce dernier a imposé une adaptation d'IPSec à l'actuel protocole IPv4. Plusieurs RFC successives décrivent les différents éléments d'IPSec : RFC 2401, 2402, 2406, 2408…

On établit un tunnel entre deux sites (voir figure 10.6), et IPSec gère l'ensemble des paramètres de sécurité associés à la communication. Deux machines passerelles, situées à chaque extrémité du tunnel, négocient les conditions de l'échange des informations : quels algorithmes de chiffrement, quelles méthodes de signature numérique ainsi que les clés utilisées pour ces mécanismes. La protection est apportée à tous les trafics et elle est transparente aux différentes applications.

Figure 10.6

Un tunnel IPSec entre deux sites d'entreprise.



IPSec prévoit la définition de la politique de sécurité avec le choix des algorithmes utilisés et leur portée. Une fois qu'une politique est définie, il y a échange des clés avec un mécanisme IKE (*Internet Key Exchange*) [utilisant le port 500 et le transport UDP]. On peut mettre en œuvre l'authentification soit en supposant que les deux extrémités se partagent déjà un secret pour la génération de clés de sessions, soit en utilisant des certificats et des signatures RSA. Les machines passerelles traitent ensuite les données avec la politique de sécurité associée.

À titre d'exemple, il est possible d'authentifier les adresses IP utilisées ainsi que les données grâce à une signature numérique puis chiffrer l'ensemble du paquet IP en l'encapsulant dans un nouveau paquet. Cela a pour effet de rendre le paquet inexploitable par un utilisateur non autorisé. En effet, IPSec propose ensuite deux mécanismes au choix pour les

données de l'échange : ESP (*Encapsulating Security Payload*) et AH (*Authentication Header*). ESP fournit l'intégrité et la confidentialité, AH ne fournit que l'intégrité.

Les adresses IP des datagrammes qui circulent dans le tunnel sont celles des machines passerelles d'extrémité du tunnel. Un datagramme IPSec encapsule celui des utilisateurs d'un site à l'autre. Il est ainsi impossible de connaître les adresses IP internes en espionnant le trafic sur Internet. L'intérêt de la solution des tunnels IPSec réside dans le fait que les utilisateurs ne voient rien, aucun logiciel n'est nécessaire sur leurs machines. Cependant pour des utilisateurs mobiles, il faut envisager une autre configuration puisque leur trafic ne passe pas par la machine passerelle. Un mode IPSec dit *transport* répond à cette situation. À la différence du mode tunnel, il nécessite d'installer un logiciel spécifique sur chaque poste client pour gérer les paramètres de sécurité, faire le chiffrement et calculer les signatures.

TLS (*Transport Layer Security*)

Placer la sécurité au niveau de la couche Transport (ou à un niveau intermédiaire entre le transport et les applications[4]) pour des communications sur Internet, telle était l'idée de Netscape en 1995, quand il a proposé un paquetage de sécurité SSL (*Secure Socket Layer*) dans son célèbre navigateur. SSL construit une connexion sécurisée entre deux sockets (d'où son nom) avec négociation de paramètres et authentification mutuelle des deux extrémités.

SSL est utilisé, par exemple, pour les paiements électroniques en ligne. Une session chiffrée permet l'envoi du numéro de carte bancaire. Le chiffrement utilise l'algorithme RSA pour l'authentification et un algorithme symétrique (DES, IDEA, 3DES…) pour garantir la confidentialité de la transmission. On y ajoute une fonction de hachage comme MD5 pour assurer l'intégrité de la transmission.

SSL a connu un énorme succès. La plupart des navigateurs Web ont par la suite intégré SSLv2 (version 2) puis SSLv3 (version 3) et aujourd'hui TLSv1 (*Transport Layer Security*, version 1). En effet, l'IETF (*Internet Engineering Task Force*) a standardisé la version 3.1 et l'a rebaptisée TLS dans la RFC 2246. Quand une transaction est sécurisée, les navigateurs affichent généralement un cadenas fermé dans un coin de l'écran.

Dans la pile de protocole TCP/IP, SSL se situe entre les couches Applications et la couche Transport TCP. SSL fonctionne de manière indépendante par rapport aux applications qui l'utilisent. Son utilisation la plus courante est avec HTTP, le protocole de transport des données des pages Web. L'utilisateur choisit dans son navigateur d'employer la navigation classique (HTTP) ou la navigation sécurisée (HTTPS), ce qui se lit dans l'URL affichée. HTTPS signifie en fait HTTP avec mécanismes de sécurité SSL. L'ICANN a affecté un nouveau numéro de port pour cette application, le port 443.

Les échanges définis par le protocole SSL se déroulent en deux phases successives, l'authentification du serveur puis celle du client. Recevant une requête d'un client, le serveur envoie son certificat au client et la liste des algorithmes qu'il peut utiliser. Le client vérifie la validité du certificat à l'aide de la clé publique de l'autorité de certification contenue dans le navigateur. Si le certificat est valide, le client génère une première clé qu'il envoie au serveur chiffrée avec la clé publique du serveur. Cette clé servira à calculer une clé de session pour les données échangées par la suite entre le client et le serveur. Le serveur peut demander au client de s'authentifier (cette authentification du client est facultative). Le client répond en envoyant son certificat et des informations sur la session et le

4. On voit resurgir ici l'idée de couches session ou présentation, telles que les voyait l'ISO.

contenu des échanges précédents qu'il signe avec sa clé privée. Le serveur pourra vérifier qu'il s'agit bien du client prétendu en utilisant la clé publique de ce dernier.

Comparaison

L'intérêt principal des réseaux privés virtuels IPSec utilisant le mode tunnel est qu'ils sécurisent la totalité des flux entre les sites de l'entreprise et que les utilisateurs n'ont pas à se préoccuper de la sécurité. Tous les mécanismes sont pris en charge par les machines passerelles. Cela peut toutefois induire une lenteur dans les communications. L'intégration d'utilisateurs mobiles nécessite l'utilisation du mode transport et l'installation sur leur poste d'un logiciel spécifique pour gérer l'ensemble des mécanismes de chiffrement et de signature.

SSL permet de construire des réseaux privés virtuels dans la mesure où il sécurise les communications entre deux utilisateurs. Il offre une beaucoup plus grande souplesse puisqu'il est implémenté dans les navigateurs. Dès lors que toutes les communications de l'entreprise sont *webisées*, il est possible d'obtenir pour les trafics concernés la mise en œuvre des procédures de sécurité. De fait, toutes les applications peuvent avoir une version sécurisée : SMPTS, POP3S, IMAPS (messagerie sécurisée sur les ports 465 et 995), LDAPS (annuaire sécurisé sur le port 636)…

Les algorithmes utilisés par SSL sont vulnérables aux attaques par force brute si les clés sont courtes (40 bits). Il est donc conseillé d'utiliser des clés d'au moins 128 bits. Le protocole utilisé présente des faiblesses si un attaquant intercepte la requête du client et se fait passer pour le serveur auprès de lui, et pour un client auprès du serveur légitime. Enfin, il est rare que le client vérifie la signature du certificat, et encore plus rare qu'il consulte la liste des certificats révoqués…

3 Sécurité dans les réseaux sans fil

Un réseau sans fil non sécurisé laisse des personnes non autorisées écouter ce réseau, y accéder, voire le modifier. On peut le sécuriser de façon plus ou moins stricte à différents niveaux : configuration des équipements d'accès et choix des protocoles.

Modifier la configuration par défaut des points d'accès est la première étape. Il est conseillé de changer les mots de passe par défaut, de changer l'identificateur SSID (*Service Set IDentifier*) par défaut, de désactiver les services non utilisés et de régler la puissance d'émission au minimum nécessaire pour couvrir la zone géographique voulue. L'activation des options de sécurité, de la journalisation de l'activité, est également nécessaire. Enfin, il est important de prendre en compte la sécurité physique des points d'accès.

La deuxième étape concerne les protocoles utilisés et les mécanismes de sécurité mis en œuvre.

Toute personne munie d'une carte Wi-Fi et située dans le périmètre de réception des ondes émises reçoit l'ensemble des données qui transitent sur ce réseau. Si celles-ci ne sont pas chiffrées, il peut les exploiter à souhait. WEP (*Wired Equivalent Privacy*) définit le chiffrement entre éléments d'un réseau sans fil. Il utilise l'algorithme RC4 et nécessite une clé secrète partagée de 40 ou 104 bits. Malheureusement, il est possible de casser une clé WEP si on dispose d'une quantité suffisante de trafic et avec un ordinateur banal (en quelques minutes et avec des programmes libres téléchargeables sur Internet).

De plus, tous les utilisateurs d'un réseau protégé partagent la même clé WEP. Chacun d'eux peut écouter les autres et déchiffrer : il faudra changer la clé à chaque départ d'un utilisateur.

Le standard IEEE 802.11i introduit le protocole WPA (*Wifi Protected Access*) qui pallie les faiblesses précédentes. WPA propose un chiffrement par paquet RC4 utilisé avec des clés de 128 bits et surtout un changement automatique des clés. Il repose sur l'utilisation d'un serveur RADIUS pour l'authentification et apporte une meilleure sécurité que WEP. Une version récente intègre l'algorithme AES (*Advanced Encryption Standard*) et d'autres méthodes de chiffrement comme WRAP (*Wireless Robust Authenticated Protocol*).

Enfin, la plupart des équipements donnent la possibilité de filtrer les adresses MAC des stations du réseau. Ce mécanisme d'authentification est inefficace s'il est utilisé seul. En effet, les systèmes d'exploitation actuels permettent à un utilisateur mal intentionné de modifier son adresse MAC et d'en usurper une valide.

Le standard IEEE 802.1x, utilisable quel que soit l'environnement (sans fil ou filaire), définit l'encapsulation d'un nouveau mécanisme d'authentification EAP (*Extensible Authentication Protocol*) au-dessus de IEEE 802.11. Cette solution EAP associée à TLS s'impose aujourd'hui par sa robustesse.

Résumé

Au cours de ce chapitre, nous avons abordé les différents aspects liés à la sécurité dans les réseaux et les particularités des réseaux sans fil et des usagers nomades. L'ISO a défini le vocabulaire des services et des mécanismes de sécurité : l'authentification, l'intégrité, la non-répudiation, etc. Les solutions retenues actuellement pour faire face aux différents risques et menaces foisonnent. À titre d'exemple, nous avons décrit le chiffrement, la signature numérique, les certificats, les réseaux privés virtuels, les pare-feu, etc. La panoplie des protections est très vaste, elle s'accroît avec la créativité des attaquants ; par ailleurs, la technologie évolue et leur fournit des capacités de traitement toujours plus puissantes. La sécurité du système d'information et des réseaux nécessite donc des équipes compétentes et rigoureuses et une bonne information des utilisateurs.

Problèmes et exercices

Exercice 1 Code de César

Énoncé

Pour protéger des données confidentielles, on utilise un système de chiffrement dit de César (qui consiste à décaler les lettres de l'alphabet d'une constante). Montrer qu'il est très aisé de déchiffrer le message suivant (écrit en français) : `Zsgashwsfrwbhsfbsh`.

Solution

Le message écrit est : `LesmetiersdInternet`

Le code est le suivant : décalage de 14 lettres.

Clair : `abcdefghijklmnopqrstuvwxyz`

Chiffré : `opqrstuvwxyzabcdefghijklmn`

En français, la lettre la plus fréquente est le e : ici il y a 5 `s` et 3 `h`. Il est logique de tester `y = e`. Le reste vient tout seul ensuite puisque le décalage est constant… Le système est donc très facilement cassable dès lors qu'on connaît les fréquences des lettres dans la langue utilisée !

Exercice 2 Cryptanalyse

Énoncé

On considère un système de chiffrement symétrique avec une clé de 64 bits. Vous cherchez à casser le système sans aucune connaissance de la clé : vous essayez de manière exhaustive toutes les clés. On suppose que vous avez à votre disposition un ordinateur puissant capable de tester une clé (et de dire si c'est la bonne !) en une picoseconde.

a Combien de clés y a-t-il ? Combien de clés en moyenne essaierez-vous ?

b Combien de temps en moyenne vous faudra-t-il pour trouver la bonne clé ?

c Quelles solutions préconisez-vous pour lutter contre la cryptanalyse par force brute ?

Solution

a Il y a 2^{64} clés possibles. En moyenne, vous en essaierez la moitié (une seule avec beaucoup de chance et toutes avec beaucoup de malchance, ce qui fait en moyenne la moitié) soit 2^{63}.

b Une picoseconde = 10^{-12} s. Le temps moyen nécessaire est donc de : $263*10^{-12}$ s.

En utilisant $10^3 = 2^{10}$, nous obtenons
$2^{63}*10^{-12}$ s = 2^{23} s = 8 388 608 s = 2 330 h = 97 jours = 3 mois.

c Vous améliorerez vos « performances » avec une puissance de calcul plus grande (1 000 fois plus grande par exemple, et le temps moyen devient deux heures). Deux solutions sont possibles pour vous rendre la tâche impossible : augmenter la longueur de la clé (avec une clé de 128 bits et la puissance de calcul un million de fois plus grande qu'à la question précédente, il vous faudra tout de même six milliards de milliards de siècles… en moyenne !) et changer la clé régulièrement (si on garde les valeurs de la première question avec un temps moyen de trois mois, il peut être judicieux de changer la clé toutes les semaines !).

Exercice 3 : Cassez un système !

Énoncé

Vous avez intercepté le message suivant :

```
KAZUIVZYTJZXFPDIFFJCZQXWQZXQHRJYRHCOEKXIJZXLBVSNQTMQSYDTMSWJIH
TOSCUWRCYQQOTNCZHAVGYRBIQALTIFIDGMUAHG
```

Vous cherchez à le déchiffrer. Votre indice : il s'agit d'un code de substitution et la clé est de longueur 5.

Solution

Dans un code de substitution, la fréquence des lettres codées est la même que la fréquence des lettres dans la langue utilisée. Si vous savez que la clé est de longueur 5, il faut découper le message en blocs de 5 caractères :

```
KAZUI VZYTJ ZXFPD IFFJC ZQXWQ ZXQHR JYRHC OEKXI JZXLB VSNQT MQSYD TMSWJ
IHTOS CUWRC YQQOT NCZHA VGYRB IQALT IFIDG MUAHG
```

Il faut ensuite traiter tous les premiers caractères de blocs, tous les deuxièmes, etc. Vous obtenez, par exemple, pour tous les premiers caractères : `KVZIZZJOJVMTICYNVIIM`

Ces données font apparaître 4 I et 3 Z. On peut penser que l'une de ces deux lettres représente le E, lettre la plus fréquente en français.

Il faut faire de même avec les deuxièmes caractères : `AZXFQXYEZSQMHUWQCGQFU`

Ces données font apparaître 4 Q et jamais plus de deux fois une autre lettre. Il y a de bonnes chances pour que le Q code le E.

Les troisièmes lettres donnent : `zyffxqrkxnsstwqzyaia`

Ici, rien de significatif : 2 F, 2 Q, 2 S, 2 A, 2 Z…

Les quatrièmes lettres donnent : `UTPJWHHXLQYWOROHRLDH`

Ces données font apparaître 4 H et jamais plus de deux fois une autre lettre. Il y a de bonnes chances pour que le H code le E.

Les cinquièmes lettres donnent : `IJDCQRCIBTDJSCTABTGG`

Ici encore rien de vraiment significatif : 3 C, 2 I, 2 J, 2 B, 2 G…

Utilisons l'hypothèse la plus vraisemblable : le E est codé par Q en tant que troisième lettre et H en tant que quatrième.

Le message devient alors, en décodant toutes les deuxièmes et quatrièmes lettres :

```
.O.R. .N.Q. .L.M. .T.G. .E.T. .L.E. .M.E. .P.U. .N.I. .G.N.
.E.V. .A.T. .V.L. .I.O. .E.L. .Q.E. .U O. .E.I. .T.A. .I.E.
```

Si cette hypothèse est la bonne, on peut profiter de la présence de la lettre Q qui est presque toujours suivie d'un U en français. Nous obtenons alors le décodage de la cinquième lettre et de la troisième :

```
.OURT .NTQU .LAMO .TAGN .ESTB .LLEC .MMEN .PEUT .NSIM .GINE .ENVO .ANTU
.VOLD .IRON .ELLE .QUEL .UTOM .EVIE .TDAR .IVER
```

Quelques essais pour la première lettre montrent que le E est codé par Z et on décode finalement :

```
POURT ANTQU ELAMO NTAGN EESTB ELLEC OMMEN TPEUT ONSIM AGINE RENVO YANTU
NVOLD HIRON DELLE SQUEL AUTOM NEVIE NTDAR RIVER
```

Soit en mettant les espaces et les accents, les paroles d'une chanson de J. Ferrat :

```
POURTANT QUE LA MONTAGNE EST BELLE COMMENT PEUT ON S'IMAGINER EN VOYANT UN
VOL D'HIRONDELLES QUE L'AUTOMNE VIENT D'ARRIVER
```

Exercice 4 : Chiffrement et conséquences des erreurs de transmission

Énoncé

On utilise un mécanisme de chiffrement par bloc de 128 bits.

a Quelle sera la conséquence d'une erreur de transmission non détectée sur un bit lors du déchiffrement ?

b Quelle sera la conséquence d'un ajout ou d'une perte d'un bit lors de la transmission ?

Solution

a Le bit erroné se trouve à l'intérieur d'un bloc de 128 bits. Celui-ci sera faux après le déchiffrement. Il y aura donc 128 bits faux en tout.

b S'il y a ajout ou perte d'un bit, la frontière des blocs de 128 bits sera affectée, et tout le message à partir du bloc où a eu lieu l'erreur sera faux.

Exercice 5 : The man in the middle

Énoncé

L'un des dangers des communications est l'écoute par un attaquant qui peut intercepter les messages entre deux correspondants, ce que les Anglo-Saxons désignent par l'expression « man in the middle » (littéralement, l'homme au milieu). Considérez le scénario ci-après. *A* et *B* qui ne se connaissent pas veulent partager un secret afin de chiffrer leurs communications futures. Ils utilisent un dispositif créé par Diffie et Hellman. On suppose que deux grands nombres, *n* et *g*, sont connus et publics. *A* choisit un nombre *x* et calcule $g^x \bmod n$ qu'il envoie à *B*, lequel, de son côté, choisit un nombre *y* et calcule $g^y \bmod n$ qu'il envoie à *A*. La théorie des nombres permet de montrer que $[g^x \bmod n]^y \bmod n = [g^y \bmod n\ ]^x \bmod n = g^{xy} \bmod n$.

a Quel est l'intérêt du système de Diffie-Hellman ?

b Imaginez que vous êtes *the man in the middle*, c'est-à-dire un attaquant *C*, capable d'intercepter la communication entre *A* et *B*. Que pouvez-vous faire ?

Solution

a Le mécanisme Diffie-Hellman est très intéressant car il permet à *A* et *B* de partager un secret ($g^{xy} \bmod n$) alors que celui-ci n'a pas été transmis. *A* et *B* ont chacun choisi de leur côté un nombre (*x* ou *y*) et l'ont transmis à l'autre sous une forme chiffrée. *A priori*, la connaissance de *g* et de *n* ne permet pas de retrouver *x* ou *y*.

b Reprenons l'échange de la question précédente avec *C* au milieu. *A* choisit un nombre *x* et calcule $g^x \bmod n$ qu'il envoie à *B*… non ! à *C*, qui intercepte le message. *C* choisit un nombre *z* et calcule $g^z \bmod n$ qu'il envoie à *B*. Celui-ci choisit un nombre *y* et calcule $g^y \bmod n$ qu'il envoie, croit-il, à *A*. En fait, *C* intercepte le message et envoie $g^z \bmod n$ à *A*.

Comme dans le scénario précédent, *A* calcule $[g^z \bmod n]^x \bmod n = g^{xz} \bmod n$, secret qu'il croit partager avec *B* alors qu'il le partage avec *C*. De même, *B* calcule $[g^z \bmod n]^y \bmod n = g^{yz} \bmod n$, secret qu'il croit partager avec *A*… Le tour est joué, *C* partage un secret avec chacun des deux correspondants et peut déchiffrer les communications, voire infiltrer des messages dans la communication ou en perdre.

Remarque

La faiblesse de ce système provient du fait qu'il n'y a aucune authentification.

EXERCICE 6 AUTHENTIFICATION

Énoncé

Supposons que deux correspondants A et B se partagent un secret K_{AB}. Ils utilisent le mécanisme d'authentification suivant : A envoie son identité accompagnée d'un nombre aléatoire N_A à B, qui renvoie en retour un nombre aléatoire N_B et le nombre envoyé par A chiffré avec la clé partagée K_{AB}. A renvoie enfin à B le nombre N_B chiffré avec la clé partagée K_{AB}.

a A et B sont-ils mutuellement certains de leurs identités respectives ?

b Imaginez un attaquant C placé entre A et B comme à l'exercice précédent, interceptant le trafic et se faisant passer pour A auprès de B et pour B auprès de A. C peut-il encore pénétrer les communications entre A et B ?

Solution

a A et B sont mutuellement authentifiés après le « défi » qu'ils se sont lancé : êtes-vous capable de chiffrer avec notre clé partagée le nombre aléatoire que je viens de vous envoyer ?

b L'attaquant C peut encore faire des ravages ! Imaginons qu'il intercepte le premier message. Il le change et envoie à B l'identité de A accompagnée d'un nombre aléatoire N_C qu'il a lui-même choisi. B envoie en retour un nombre aléatoire N_B et le nombre envoyé par C, chiffré avec la clé partagée K_{AB}. Il n'a que faire de ce dernier mais le nombre N_B est très précieux. Un peu plus tard, se faisant toujours passer pour A, C envoie à B le message initial : son identité (celle de A…) et le nombre N_A intercepté au début. B peut raisonnablement penser qu'il s'agit du début d'une nouvelle procédure d'authentification : il envoie en retour un nombre aléatoire N'_B et le nombre envoyé par C (c'est-à-dire N_A), chiffré avec la clé partagée K_{AB}. Et le tour est joué. C est maintenant en possession de N_B en clair et de N_A chiffré avec la clé partagée K_{AB}. C'est le message que B aurait dû envoyer à A lors de la requête initiale. C l'envoie donc et A pense alors qu'il discute avec B…

Remarque

Le scénario est devenu complexe, il faut de plus imaginer que B accepte deux sessions différentes avec A… Plus la protection est grande et plus l'attaquant doit faire preuve d'ingéniosité ! Une solution à ce nouveau problème pourrait être de dater les messages et contraindre l'intervalle de temps pour la réponse mais cela peut gêner les processus normaux autorisés !

EXERCICE 7 RÈGLES D'UN PARE-FEU

Énoncé

Écrivez en pseudo-langage les règles de filtrage nécessaires pour rejeter, en entrée du routeur pare-feu du réseau `195.45.3.0` (de masque `255.255.255.0`), les attaques en déni de service : inondation de requêtes de connexion TCP ou de messages ICMP avec des adresses IP usurpées.

Solution

Usurpation d'adresse : des messages proviendraient de l'extérieur du réseau avec une adresse d'émetteur qui est une adresse interne.

Définition des paramètres : `ouraddr = 195.45.3.0/24` (toutes les adresses de notre réseau) ; `anyaddr` = n'importe quelle adresse

Effacer toutes les règles en entrée ; refuser les messages en entrée dont l'adresse source est `ouraddr`, l'adresse destination est `ouraddr`, le protocole est TCP et le bit SYN est mis à 1 ;

refuser les messages en entrée dont l'adresse source est `ouraddr`, l'adresse destination est `ouraddr` et le protocole est ICMP.

EXERCICE 8 SIGNATURE PGP

Énoncé

Vous recevez le message suivant. Que vous apprend-il ? (Le corps du message a été remplacé par ////.)

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1
////////////
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.2.2 (GNU/Linux)
Comment: Using GnuPG with Fedora - http://enigmail.mozdev.org
iD8DBQFEs1EzNVD/FpaFUPQRAsDfAKCKlZHGp9HIQhpG1Ynp7CSh5E1MEgCaAzim
dG9tb1Qjge0fXA5NOLn4UsY=
=hPTs
-----END PGP SIGNATURE-----
```

Solution

Ce courrier électronique émane d'un utilisateur qui a choisi d'implémenter PGP sur sa machine (paquetage libre GnuPG). Le message est signé avec l'algorithme SHA1. Vous êtes certain que son contenu n'a pas été altéré au cours de la transmission.

EXERCICE 9 IPSEC ET NAT

Énoncé

Le mécanisme AH dans IPSec authentifie un datagramme dans son ensemble et intercale un champ supplémentaire d'authentification entre l'en-tête IP et le contenu du datagramme.

a Quels sont les champs de l'en-tête du datagramme IP qui doivent être exclus de ce mécanisme ?

b Quelle sécurité a-t-on quant aux adresses IP utilisées ?

c On utilise IPSec avec AH en mode transport pour un utilisateur mobile qui se trouve derrière un pare-feu avec mécanisme NAT. Quelle est la conséquence ?

Solution

a Les routeurs décrémentent le champ TTL de l'en-tête de chaque datagramme et refont le calcul du bloc de contrôle d'erreur de l'en-tête de ce fait. Ces deux champs ne doivent donc pas entrer dans le mécanisme d'authentification. Sinon, tous les paquets IP seraient falsifiés à la première traversée d'un routeur.

b Le mécanisme d'authentification garantit les adresses IP, il ne garantit pas les personnes qui les utilisent.

c Lorsqu'un datagramme traverse un routeur pare-feu qui utilise le mécanisme NAT, l'adresse IP interne est remplacée par une nouvelle adresse IP publique. Cela est incompatible avec le mécanisme NAT.

Remarque

Il a fallu trouver des adaptations de NAT pour contourner le problème. C'est l'une des occasions qui nous a fait qualifier de « bricolage » bien des solutions proposées.