N° 36 MARS/AVRIL 2008

France Métro : 8 € DOM : 8,80 € TOM : 990 XPF CHF 15,50
BEL, LUX, PORT.CONT : 9 Eur CAN : 15 $CAD

MISC

Multi-System & Internet Security Cookbook

100 % SÉCURITÉ INFORMATIQUE

PROGRAMMATION

Comment réaliser un fuzzer ?

Découvrez la manière d'injecter des données aléatoires dans les entrées d'un programme pour tester sa sécurité *(p. 68)*

DOSSIER

LUTTE INFORMATIQUE OFFENSIVE

LES ATTAQUES CIBLÉES

- **Identification des failles humaines pour prédateur informationnel**
- **Outil spécifique pour attaques ciblées d'entreprises**
- **Guerre de l'information sur Internet**



CRYPTOGRAPHIE

Comprendre les modes de chiffrement par blocs (p. 12)

RÉSEAU

Répartition de charges par la pratique (p. 74)

ORGANISATION

PCI-DSS : une norme de protection des données de cartes bancaires bientôt obligatoire ? (p. 4)

Guide d'hygiène informatique
Les attaques ciblées

ÉDITO ▶ Apparences trompeuses

Souvenez-vous... Pagrec, le paladin, amoureux des langues anciennes (et non pas gigolo) ; Derien, le clerc, et son air de ne pas y toucher ; Metha, le voleur, avec sa ponctualité défaillante ; Canonix, le mage, pas encore bon pour la maison de retraite.

L'an passé, nos héros étaient finalement venus à bout du terrible dragon noir et avaient récupéré le joyau impérial [1]. Depuis, tout roule ! Plébiscités par le roi, ils se voient remettre le sceptre de la Justice. Plébiscités par le jury populaire, ils sont devenus la crème des stars. La vie suit son cours, de quêtes en sauvetages du monde, la routine quoi (enfin, Pagrec commence à prendre un peu de bide, à bouffer tout le temps des sandwiches).

Pendant ce temps, dans l'ombre, la menace grandit. Cette fois, elle se présente sous la forme d'un petit groupe de super méchants : Dambert, assassin fourbe ; Tyck, nécromant poète ; Niu, Orc guerrier au séant athlétique. Depuis plusieurs semaines, ils préparent leur plan pour se débarrasser de la garde rapprochée, pour subtiliser le sceptre aux tocards. Mais pour l'instant, tintin en attendant la bonne occasion (celle qui fait le larron bien sûr).

Ils commencent à mettre en place des filatures, acheter des indicateurs, fouiller les poubelles de la maison des héros. Les cibles ont d'énormes moyens maintenant. Faut dire, un trésor de dragon avec son épée vorpale +3 à deux mains avec poignée cuir et garde amovible, son armure en cuir +5 contre les coups de soleil et son collant moule-burnes toujours en vogue à la cour et qui ne gratte pas, ça permet de voir venir. Heureusement que nos aventuriers ont à la fois un goût pour l'ancien (mais non, pas le père Fouras) et le sens des affaires.

D'ailleurs, en se promenant, Canonix rencontre par hasard au détour d'une chope de bière un jeune homme avec qui il sympathise. Il apprend ainsi que son grand-père Assou-an, antiquaire, va être expulsé de sa boutique. Elle sera vendue aux enchères, avec les vieilleries qu'elle contient, pour rembourser une créance. N'y connaissant rien, le jeune Assou-an demande à son aîné de l'accompagner et de lui prodiguer quelques conseils (oui, oui, c'est un roman, ça n'arrive jamais en vrai ça ;-)

Après quelques minutes de fouilles, Canonix découvre un amas de caisses. Celle du dessus est éventrée : un vieux morceau de tissu tout froissé dépasse. Quand il glisse sa main dessous pour fouiner, elle semble disparaître, mais comme le petit Assou-an arrive juste à ce moment-là, il laisse la fripe sur la caisse. « D'où viennent tous ces lambeaux de tissu ? » Pas un bruit dans la cave, c'est le silence des lambeaux avant la réponse d'Assou-an : « mon grand-père fut aventurier dans sa jeunesse, et lors de l'invasion rafobik, il participa au célèbre barrage qui les empêcha de remonter ». La visite de l'échoppe continue et Assou-an s'aperçoit qu'il joue machinalement avec le bout de tissu. Il le jette par la fenêtre sous les yeux exorbités de Canonix, qui pense tenir là une cape d'invisibilité et l'inestimable trésor rafobik !

Canonix retourne voir ses potes et leur propose de racheter le petit commerce. Sous réserve que la cape soit authentique... ce qui s'avère être le cas. Le lendemain, nos 4 amis participent aux enchères. Manque de bol, un vieux croûton dans un coin se manifeste aussi pour acquérir la maison. Il semble que ce soit un autre antiquaire, Puh-Blik, fameux sur la place. Du coup, les offres s'envolent. Nos aventuriers, plus habitués aux marteaux des nains qu'à ceux des commissaires-priseurs, se consultent. Pour suivre et remporter l'enchère, ils doivent rassembler toutes leurs économies. Banco ! Et dans une association unanime, synonyme de société générale, ils emportent la mise grâce à la dernière surenchère.

Les choses ne sont pas toujours ce qu'elles paraissent. D'un côté, tout le monde clame – à juste titre – que les méchants sont de plus en plus organisés, compétents, imaginatifs. On a ainsi vu l'impact de Storm ou de MPack, kits de malfaiteurs prêts à l'emploi. Côté terroriste, après le magazine (le *Technical mujahideen* vient l'heure du logiciel de chiffrement (*Mujahideen secrets*). Sans oublier, au milieu de cette cour maléfique, le pirate qui télécharge de la musique et des films (bouh, honte à lui !).

En face, on a les gentils. Pour lutter, ils mettent en place un site web destiné à sensibiliser le public [2] ou proposent des lois pour pirater (oups, ah non, pardon, ce sera sous le contrôle d'un juge, donc ce sera légal) les ordinateurs des suspects à distance. Ou alors, ils payent (certains journalistes étrangers parlent de corruption, mais ce n'est après tout qu'une compensation pour service rendu) un informateur pour récupérer des listings de comptes (tiens, ça me rappelle une autre histoire ça !). Autre innovation à l'étude, permettre aux représentants des majors de venir se brancher directement sur les équipements des FAI. Remarque, un certain opérateur historique a trouvé la parade en délocalisant en Égypte toute l'administration de son infrastructure (si, si !).

Revenons sur les événements. En réalité, en préparant l'opération, Dambert le fourbe a découvert que nos héros profitaient de leur position de gardiens du sceptre : lors des séances publiques, l'équité était faussée par des « offrandes aux héros » distribuées discrètement juste avant le début de la cérémonie. Du coup, profitant de l'avidité des 4, ils leur font miroiter le magot rafobik par l'unique pièce authentique qu'ils ont acquise par ailleurs. Dambert se grime en Assou-an. Canonix et ses amis, plein d'avides espoirs, sont ruinés aux enchères face à un faux Puh-Blik (Tyck déguisé). Ils quittent la ville qui retrouve ainsi la Justice.

Bref, je ne sais pas si la fin justifie les moyens, mais je me pose quand même des questions sur lesdits « moyens »... En tout cas, les méchants ne se posent pas trop de questions quant aux retours sur investissement, alors que les gentils égorgent toutes les chèvres qu'ils peuvent pour que leur sécurité leur coûte le moins cher possible. D'un côté, il y en a qui se bougent pour arriver à leurs fins, de l'autre, on copie les méchants à retardement. Asymétries problématiques...

Bonne lecture,

Fred Raynal

P.S. : quelques messages plus ou moins personnels pour conclure :

- ➪ Bon 86ème anniversaire à ma grand-mère :)
- ➪ Le programme du SSTIC est en ligne [3], les places devraient bientôt suivre (si ce n'est déjà le cas quand vous lirez cet édito).
- ➪ Bon courage à Cédric 0x90 Bubu et Sophie Python Scapy (pas des prénoms faciles) pour supporter leurs nouveaux parents.

[1] http://miscmag.com/fr/index.php?2007/03/04/28-livre-dont-vous-etes-le-heros

[2] http://www.securite-informatique.gouv.fr/

[3] http://www.sstic.org/SSTIC08/programme.do

SOMMAIRE

ORGANISATION [04 - 10]

> PCI-DSS : une norme de protection des données de cartes bancaires bientôt obligatoire ?

CRYPTOGRAPHIE [12 - 21]

> Les modes de chiffrements par blocs

DOSSIER [22 - 67]
[Lutte informatique offensive : les attaques ciblées]

> Les acteurs de la LIO : les « bons », les « brutes » et les « truands »... / 22 → 31

> Petit mémo d'identification des failles à l'usage du prédateur informationnel... / 32 → 44

> Outil spécifique pour attaques ciblées d'entreprises (partie 1) / 47 → 55

> Petit traité d'e-manipulation à l'usage des honnêtes gens (ou pas) / 56 → 67

PROGRAMMATION [68 - 73]

> Comment réaliser un fuzzer ?

RÉSEAU [74 - 82]

> Répartition de charges par la pratique (partie 1)

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 :
cial@ed-diamond.com

Service commercial :
abo@ed-diamond.com

Sites :
www.ed-diamond.com

www.miscmag.com

LES ÉDITIONS DIAMOND

Printed in France
Imprimé en France
Dépôt légal : 2e Trimestre 2001
N° ISSN : 1631-9036
Commission Paritaire : 02 09 K80 190
Périodicité : Bimestrielle
Prix de vente : 8 Euros

Directeur de publication : Arnaud Metzler
Chef des rédactions : Denis Bodor
Rédacteur en chef : Frédéric Raynal
Relecture : Dominique Grosse
Secrétaire de rédaction: Véronique Wilhelm
Conception graphique : Kathrin Troeger
Responsable publicité : Tél. : 03 88 58 02 08
Service abonnement : Tél. : 03 88 58 02 08
Impression : I.D.S. Impression (Sélestat) / www.ids-impression.fr
Distribution France : (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

La rédaction n'est pas responsable des textes, illustrations et photos qui lui sont communiqués par leurs auteurs. La reproduction totale ou partielle des articles publiés dans Misc est interdite sans accord écrit de la société Diamond Editions. Sauf accord particulier, les manuscrits, photos et dessins adressés à Misc, publiés ou non, ne sont ni rendus, ni renvoyés. Les indications de prix et d'adresses figurant dans les pages rédactionnelles sont données à titre d'information, sans aucun but publicitaire.

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 ses 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.

Sylvain ROGER – Responsable Offre Audit, Solucom Group – sylvain.roger@solucom.fr

PCI-DSS : UNE NORME DE PROTECTION DES DONNÉES DE CARTES BANCAIRES BIENTÔT OBLIGATOIRE ?

Un consortium regroupant les plus importantes sociétés de l'industrie des cartes de paiement (Visa, MasterCard, American Express…) a défini un standard de protection des données de cartes : le PCI-DSS (Payment Card Industry Data Security Standards). Après les États-Unis, ce standard risque fort de devenir obligatoire en Europe. Regardons d'un peu plus près les exigences demandées et les conséquences concrètes pour l'ensemble des acteurs du cycle de paiement par carte.

mots clés : *norme / cartes bancaires / audit / conformité*

1. Quelques éléments de contexte

1.1 Un moyen de paiement de plus en plus utilisé

Selon une étude Sofres, il y avait 53 millions de cartes bancaires en France en 2006, réparties presque exclusivement entre Visa et MasterCard :

Introduite dans les années 50, l'importance de la carte bancaire en France et dans le monde n'a cessé d'augmenter. Selon Visa, la part des dépenses des ménages français en valeur effectuée par Carte Bleue est passée de 12% en 2005 à 16% en 2006.

Par ailleurs, le montant des fraudes pour les transactions domestiques était de 109,6 millions d'euros en 2006, soit 0,031%

	Visa [1]	MasterCard [2]
Nombre de cartes en France	31,88 millions	21 millions
Nombre de transactions domestiques (et évolution)	4,15 milliards (+7,5%)	2,5 milliards (+8,5%)
Montant des transactions domestiques (et évolution)	219,1 milliards € (+8,1%)	127,4 milliards € (+10%)

du montant total des transactions, selon l'observatoire de la sécurité des cartes de paiement [3]. Si ce pourcentage peut paraître dérisoire, l'impact en termes d'image peut être conséquent, et de nombreuses personnes hésitent encore souvent à faire leurs achats sur Internet par peur des fraudes. Pourtant, si, dans l'esprit des gens, la fraude au niveau des Cartes Bleues provient d'Internet, la réalité est tout autre : la fraude sur internet ne représenterait, en termes de montant, pas plus de 1,25% des fraudes.

Pour pallier ces états de fait, les sociétés de cartes bancaires ont donc décidé d'uniformiser et de renforcer leurs politiques en matière de protection des données stockées sur les cartes bancaires.

1.2 Quelles informations dans une carte bancaire ?

Les informations qui sont à protéger au niveau d'une carte bancaire sont, pour rappel, les suivantes :

1. Le **Primary Account Number** (PAN) : le PAN est le numéro inscrit « en dur » sur la carte bancaire. Il est composé d'un préfixe et d'un suffixe. Le préfixe, de 6 chiffres pour les cartes VISA et MasterCard, identifie la banque émettrice. Le suffixe, généralement de 9 chiffres, est un numéro individuel. Enfin, le dernier chiffre est un code de vérification, permettant, par exemple, d'éviter les erreurs de saisie lors d'un achat sur Internet. Le stockage, la transmission, et le traitement du PAN sont autorisés, mais ils entraînent, pour l'entreprise concernée, une obligation de se conformer aux PCI-DSS. Inversement, une entreprise qui ne stocke, ne transmet, ni ne traite le PAN n'est pas soumise aux PCI-DSS.

> *...l'impact en termes d'image peut être conséquent...*

2. Le **nom du titulaire** et la **date d'expiration** : le stockage, la transmission, et le traitement du nom du titulaire et de la date d'expiration sont autorisés, et ils n'entraînent, à eux seuls, aucune obligation de conformité aux PCI-DSS.

3. **Le code de service** : le code de service est une information qui est décrite, dans les documentations disponibles sur le site du PCI SSC, comme autorisée au stockage, au traitement, et à la transmission, et n'entraînant à elle seule aucune obligation de conformité aux PCI-DSS.

4. **La bande magnétique, le CVV et le code PIN** : toutes les informations bancaires sont présentes dans la bande magnétique. Le CVV (*Card Verification Value*) ou CVC (*Card Verification Code*) est le numéro, inscrit au dos de la carte, qui permet d'augmenter la sécurité des transactions par Internet et par téléphone. Le code PIN ou *Personnal Identification Number* permet de vérifier simplement, sans avoir à consulter la carte d'identité, que la personne qui utilise une carte bancaire est bien son détenteur légitime.

1.3 De multiples acteurs dans le cycle de paiement par carte

1.3.1 Quelques définitions préalables

Les commerçants : les commerçants sont bien évidemment particulièrement concernés par les PCI-DSS. En effet, de nombreux petits commerçants n'ont pas les moyens, l'infrastructure ou la culture nécessaires pour adopter de bonnes pratiques en matière de protection des données des cartes bancaires, et mettent donc potentiellement en péril ces données.

Les sociétés de cartes bancaires : les sociétés de cartes bancaires, telles Visa, MasterCard et American Express, sont des sociétés financières qui ont été à l'origine de la création des cartes bancaires. Ces sociétés traitent les transactions bancaires faites à partir des cartes bancaires. Elles servent d'intermédiaires entre les banques des commerçants et les banques des clients.

Les acquéreurs : les acquéreurs, ou banques acquéreurs, sont des banques qui permettent aux commerçants d'accepter les cartes bancaires de la part de leurs clients. Les acquéreurs vendent ou louent des appareils permettant la lecture des cartes, et se chargent d'effectuer les transactions bancaires via les sociétés de cartes bancaires.

Les émetteurs : les émetteurs sont des banques qui permettent, entre autres aux particuliers, de posséder une carte bancaire.

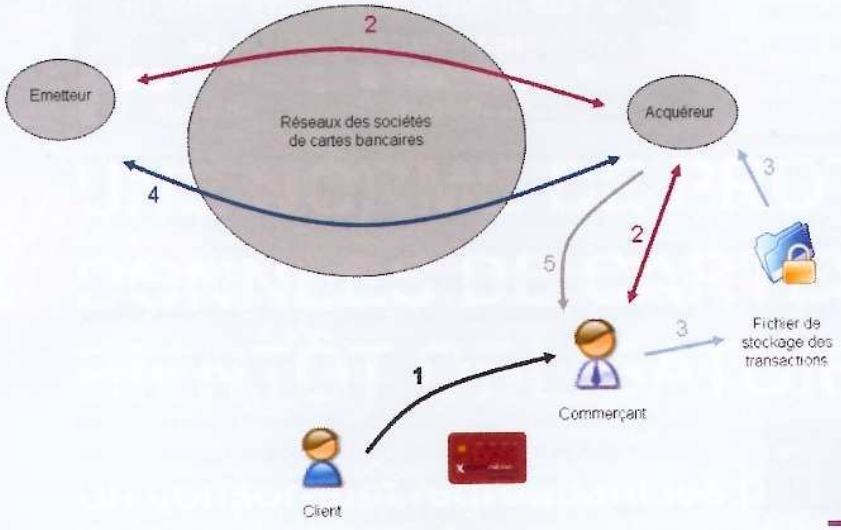
Les prestataires de services : les prestataires de services sont toutes les entreprises qui stockent, transmettent ou traitent les données des cartes bancaires (hébergeurs, fournisseurs d'accès internet, éditeurs de logiciels de comptabilité, etc.).

1.3.2 Le cycle de paiement par carte

Le processus de paiement n'est pas le seul processus à nécessiter une protection des données des cartes bancaires : les retraits, consultation de comptes, et autres transactions bancaires sont également concernées. Cependant, la complexité du processus de paiement mérite un éclaircissement. Son explication permet par ailleurs de mieux cerner les relations entre les différents acteurs engagés dans les transactions bancaires, et de comprendre pourquoi la formation du PCI SSC est le fait des sociétés de cartes bancaires.

1. Lorsqu'un particulier achète un produit chez un commerçant, il fournit au commerçant sa carte bancaire, et, selon les cas, son code PIN, sa carte d'identité ou sa signature.

2. Le commerçant demande à son acquéreur de vérifier que le client est en droit d'effectuer la transaction. L'acquéreur du commerçant vérifie la validité de la transaction avec l'émetteur du client, en utilisant les réseaux fournis par les sociétés de cartes bancaires (Visa, MasterCard, etc.). Cette étape est presque instantanée.



3▸ Une fois la transaction validée, elle est stockée dans un fichier, que le commerçant envoie à son acquéreur ultérieurement, à la fin de chaque journée par exemple.

4▸ L'acquéreur envoie les transactions aux réseaux des sociétés de cartes bancaires qui s'occupent de débiter la banque émettrice, et de créditer l'acquéreur.

5▸ Une fois que l'acquéreur a été crédité, le commerçant l'est à son tour. Il est crédité du montant de l'achat diminué de charges ponctionnées par l'acquéreur, l'émetteur, et le réseau de cartes bancaires.

À noter cependant la possibilité pour le commerçant d'externaliser complètement la gestion des transactions à un acteur spécialisé.

2. PCI-DSS : une présentation synthétique

2.1 Les acteurs du standard PCI-DSS

Le regroupement de cinq sociétés importantes de cartes bancaires a abouti à la formation du PCI SSC [5] (*Payment Card Industry Security Standards Council*), qui est en charge de la gestion des PCI-DSS. Ces « standards » ont pour but la protection des données des cartes bancaires. Ses diverses missions sont :

1▸ Développer, améliorer et publier les PCI-DSS.

2▸ Former, certifier et réévaluer régulièrement les organismes certificateurs de la conformité aux PCI-DSS.

3▸ Organiser un lieu de rencontre où les acteurs concernés peuvent effectuer des retours et fournir des recommandations au PCI SSC quant à l'évolution des PCI-DSS.

En aucun cas, le PCI SSC n'a pour mission de certifier la conformité des entreprises aux PCI-DSS, ni d'appliquer des sanctions aux entreprises ne s'y conformant pas : ces missions sont assurées directement par les sociétés de cartes bancaires (Visa, MasterCard,...) et leurs intermédiaires.

2.2 La classification des commerçants et prestataires de service

La classification des commerçants pour Visa et pour Mastercard est fonction du nombre total de transactions que le commerce effectue avec des clients possédant une carte Visa ou MasterCard.

Les prestataires de service tout comme les commerçants sont concernés.

Niveau du commerçant	Nombre de transactions	Date limite de conformité
Niveau 1	Commerces totalisant plus de 6 millions de transactions par an avec Visa et MasterCard (toutes transactions confondues)	MasterCard : 30/06/2005 Visa : 30/09/2004
Niveau 2	Commerces totalisant entre 1 million et 6 millions de transactions par an avec Visa et MasterCard (toutes transactions confondues)	MasterCard : 31/12/2008 Visa : 30/09/2007
Niveau 3	Commerces totalisant entre 20 000 et 1 million de transaction *e-commerce* par an avec Visa et MasterCard	MasterCard : 20/06/2005 Visa : 30/06/2005
Niveau 4	Commerces totalisant moins de 20 000 transactions *e-commerce* par an, et ceux qui effectuent moins de 1 million de transactions (toutes transactions confondues) avec Visa et MasterCard	MasterCard : consulter l'acquéreur Visa : consulter l'acquéreur

Niveau du prestataire	Condition pour Visa	Condition pour MasterCard	Date limite de conformité
Niveau 1	Les entreprises gérant un point VisaNet et les passerelles de paiement.	Tous les TPP ; les DSE qui gèrent des données pour des commerçants de niveau 1 et 2	MasterCard : 30/06/2005 Visa : 30/09/2004
Niveau 2	Tous les prestataires non inclus dans le niveau 1, et qui traitent, stockent ou transmettent plus d'un million de transactions ou d'informations par an	DSE qui stockent des données pour des commerçants de niveau 3	MasterCard : 30/06/2005 Visa : 30/09/2004
Niveau 3	Tous les prestataires non inclus dans le niveau 1, et qui traitent, stockent ou transmettent moins d'un million de transactions ou d'informations par an	DSE qui ne sont pas incluses dans les niveaux 1 et 2	MasterCard : 30/06/2005 Visa : 30/09/2004

Par prestataire de service, on entend toute entreprise ou organisme qui traite, stocke ou transmet des informations de carte bancaire. MasterCard fait la différence entre les entités qui traitent (*Third Data Processors, TTP*), et les entités qui stockent les données (*Data Storage Entities, DSE*).

2.3 Les spécifications du standard

Les spécifications du standard PCI-DSS s'applique à tout composant qui traite, stocke ou transmet des données des cartes bancaires. Composant est à prendre au sens large : applications, serveurs, OS, routeurs et autres composants réseau,...

Le but ici n'est pas de vous présenter en détail les spécifications du standard, mais de citer les grandes thématiques de sécurisation abordées.

La norme en matière de protection des données est constituée de 12 exigences regroupées sous les thèmes suivants :

- **Mise en place et gestion d'un réseau sécurisé**
  - *Exigence 1* : Installer et gérer une configuration de pare-feu afin de protéger les données des titulaires de carte.
  - *Exigence 2* : Ne pas utiliser les paramètres par défaut du fournisseur pour les mots de passe et les autres paramètres de sécurité du système.
- **Protection des données des titulaires de carte**
  - *Exigence 3* : Protéger les données des titulaires de carte stockées (chiffrement entre autres).
  - *Exigence 4* : Chiffrer la transmission des données des titulaires de carte sur les réseaux publics ouverts.
- **Mise en place d'un programme de gestion des vulnérabilités**
  - *Exigence 5* : Utiliser et mettre à jour régulièrement un logiciel antivirus.
  - *Exigence 6* : Développer et gérer des applications et systèmes sécurisés.
- **Mise en œuvre de mesures de contrôle d'accès efficaces**
  - *Exigence 7* : Limiter l'accès aux données des porteurs de carte aux cas de nécessité professionnelle absolue.
  - *Exigence 8* : Attribuer une identité d'utilisateur unique à chaque personne disposant d'un accès informatique.
  - *Exigence 9* : Limiter l'accès physique aux données des titulaires de carte.
- **Surveillance continue et tests des réseaux à une fréquence régulière**
  - *Exigence 10* : Suivre et surveiller tous les accès aux ressources du réseau et aux données des titulaires de carte.
  - *Exigence 11* : Tester régulièrement les systèmes et procédures de sécurité.
- **Établissement d'une politique en matière de sécurité de l'information**
  - *Exigence 12* : Disposer d'une politique régissant la sécurité de l'information.

La plupart de ces exigences ne sont que des bonnes pratiques applicables à n'importe quel type de données sensibles à protéger. Faisons néanmoins un focus sur l'exigence 3 qui est spécifique aux données de carte bancaire : celle-ci met l'accent sur le chiffrement comme composante essentielle de la protection des données de titulaires de carte. D'autres méthodes de sécurisation sont exigées pour in fine atténuer les risques encourus : absence de stockage des données de carte de crédit à moins que cela ne soit absolument nécessaire, tronquer et/ou masquer les données du titulaire de carte si un PAN complet n'est pas nécessaire...

Le PCI-DSS exige de ne jamais stocker la totalité du contenu d'une quelconque piste de la bande magnétique. En particulier, il est **strictement interdit** de stocker le code vérification de la

carte (trigramme) et des éléments de données de valeur de vérification du code PIN. Le PAN doit être masqué lorsqu'il est affiché. Cependant, cette exigence ne s'applique pas aux employés et aux autres parties ayant spécifiquement besoin d'avoir connaissance de la totalité du PAN (par exemple pour du recouvrement ou en cas d'incident de paiement). Si le PAN doit être stocké, il doit être rendu illisible par l'intermédiaire de l'une des quatre techniques suivantes :

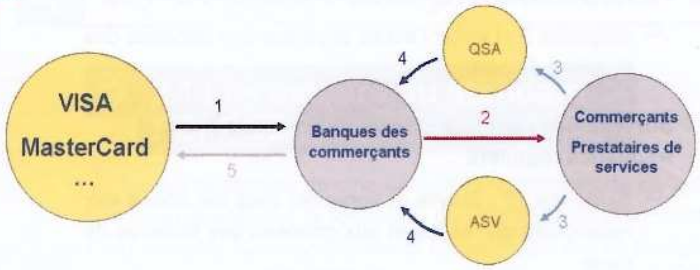
- fonctions efficaces de hachage à sens unique (aucun algorithme spécifique n'est exigé) ;
- troncature ;
- jetons et *pads* index ;
- cryptographie performante avec processus et procédures de gestion des clés associés.

Enfin, les exigences 3.5 et 3.6 adressent les problématiques de protection et de gestion des clés de chiffrement, permettant d'assurer la cohérence globale du cadre de sécurisation.

3. Quelles conséquences pour les différents acteurs ?

3.1 Obtention de la certification de conformité aux PCI-DSS

Le processus global de certification de conformité aux PCI-DSS est le suivant :



1. Demande mensuelle ou trimestrielle de rapports de conformité par les sociétés de cartes bancaires.
2. « Pression » sur les commerçants et les prestataires de service de la part des banques des commerçants.
3. Demande d'audits et de scans de vulnérabilité à des prestataires agréés.
4. Mutualisation par les banques des résultats des évaluations de conformité fournis par les évaluateurs suite aux audits et scans.
5. Communication des rapports listant les commerçants, les prestataires de services et leurs niveaux de conformité aux sociétés de cartes bancaires.

Pour les commerçants, les conditions pour être en accord avec les programmes de protection des données bancaires sont uniformisées.

L'audit annuel doit être un audit réalisé par un QSA (*Qualified Security Assessors*). Le scan de vulnérabilités trimestriel doit être effectué par un ASV (*Approved Scanning Vendors*). Pour être en conformité avec les programmes de protection de Visa et Mastercard, les commerçants doivent s'assurer que tous les prestataires de services qui traitent, transmettent ou stockent des données de cartes bancaires en leur nom sont en conformité avec les PCI-DSS.

Niveau du commerçant	Conditions		
	Audit annuel sur site	Scan de vulnérabilités trimestriel	Questionnaire d'auto-évaluation annuel
Niveau 1	Oui	Oui	Non
Niveau 2	Non	Oui	Oui
Niveau 3	Non	Oui	Oui
Niveau 4	Non	Oui	Oui

3.1.1 Les « Qualified Security Assessors » (QSA)

Les QSA sont des organismes certificateurs de la conformité aux PCI-DSS dont la formation et la certification sont assurées par le PCI SSC. Les QSA effectuent des audits de sécurité dans les entreprises, et rédigent des RoC (*Report of Compliance*) qu'ils envoient aux acquéreurs. Les QSA fournissent des documents relatifs à la conformité des entreprises aux PCI-DSS, mais c'est aux acquéreurs et aux sociétés de cartes bancaires que revient le dernier mot : sur la base des documents fournis par les QSA, ce sont eux qui décident si l'entreprise concernée remplit les conditions nécessaires.

3.1.2 Les « Approved Scanning Vendors » (ASV)

Les ASV sont des organismes certificateurs de la conformité aux PCI-DSS dont leurs formations et leurs

certifications sont assurées par le PCI SSC. Les ASV effectuent pour les entreprises des scans de vulnérabilités et rédigent des Rapports de Scan qu'ils fournissent aux acquéreurs. Les ASV fournissent des documents relatifs à la conformité des entreprises aux PCI-DSS, mais c'est aux établissements bancaires et aux sociétés de cartes bancaires que revient le dernier mot : sur la base des documents fournis par les ASV, ce sont eux qui décident si l'entreprise concernée remplit les conditions nécessaires.

3.2 Les moyens de pression de Visa et MasterCard

Visa et MasterCard ont chacun défini leurs propres moyens de pression :

1. VISA via son programme « PCI CAP » : Visa est le premier à avoir commencé un programme qui allie amendes et encouragements pour accélérer la mise en conformité des acteurs aux PCI-DSS. Visa prévoit de récompenser les acquéreurs pour chaque commerçant de niveau 1 et 2 en conformité avec les PCI-DSS, à condition que tous les commerçants de niveau 1 et 2 de l'acquéreur soient conformes avant fin mars 2007. Une récompense minorée est prévue pour les acquéreurs dont les commerçants de niveau 1 et 2 se mettraient en conformité avec les PCI-DSS entre fin mars 2007 et fin août 2007. En 2006, le montant des amendes infligées aux acquéreurs par Visa est de 4,6 millions de dollars, contre 3,4 en 2005. Sur son site internet, VISA annonce qu'il peut, lors d'une fuite d'information ou lors d'une non-conformité aux PCI-DSS, infliger des amendes à l'acquéreur avec qui le commerçant est lié, mais aussi imposer des restrictions au commerçant ou à l'acquéreur. Inversement, lors d'un problème de sécurité concernant des données bancaires, si un commerçant ou un acquéreur est entièrement en conformité avec le programme SDP, et donc avec les PCI-DSS, il ne recevra pas d'amende de la part de VISA.

2. Mastercard via son programme « SDP » : sur son site internet consacré au programme SDP, MasterCard annonce qu'il pourra infliger des amendes aux acquéreurs dont les commerçants, et les prestataires de services qui travaillent avec ces commerçants, ne sont pas tous en conformité avec les PCI-DSS. Par contre, aucun montant n'est à l'heure actuelle annoncé concernant ces amendes.

3.3 Conséquences concrètes pour les entreprises

Les conséquences concrètes pour les entreprises (américaines pour le moment, sûrement européennes à moyen terme) sont importantes et nécessitent de nombreux investissements.

Pour preuve : de 2006 à 2007, le pourcentage de commerçants de niveau 1 en conformité avec les PCI-DSS est certes passé de 18% à 35%, mais ce dernier chiffre prouve que l'application des PCI-DSS n'est pas une chose simple.

...le PAN est utilisé dans beaucoup d'applications comme un identifiant...

Par ailleurs, de nombreux retours d'expérience issus d'audits menés peuvent montrer que le PAN est utilisé dans beaucoup d'applications comme un identifiant et donc que les exigences relatives à son masquage ne sont que très rarement mises en œuvre. Par ailleurs, il n'est pas rare de voir le trigramme stocké de manière temporaire dans des bases de données clients...

En France, nous pouvons considérer que le niveau de conformité actuel est particulièrement bas.

Rappelons enfin que les commerçants et les prestataires de services sont classés en quatre catégories, en fonction du nombre de transactions par carte bancaire qu'ils traitent, transmettent ou stockent. Plus le nombre de transactions est grand, plus les exigences des sociétés de cartes bancaires sont importantes. En effet, on ne peut avoir les mêmes demandes en matière de protection des données avec une petite entreprise qu'avec une multinationale. Seulement, si les commerçants de niveau 4 ne réalisent pas beaucoup de transactions comparé aux commerçants de niveau 1, ils représentent 99% du parc de commerçants. Afin de limiter les risques en rapport avec les petites et nombreuses entreprises, les sociétés de cartes bancaires s'autorisent à imposer plus de contraintes aux entreprises où des fuites de données bancaires sont advenues, c'est-à-dire à les monter de niveau.

Conclusion

Même si de nombreux évènements, en particulier aux États-Unis, relatifs à des pertes de données et en particulier à des données de type carte bancaire sont remontés ces dernières années dans la presse, l'effort de sécurisation est encore énorme. En France en particulier, de nombreuses entreprises et commerçants doivent prendre conscience des impacts que pourrait avoir l'application obligatoire du PCI-DSS. Face à cette perspective, il est fortement conseillé de s'attaquer dès aujourd'hui au sujet en effectuant a minima une mesure d'écart et une évaluation des investissements potentiels pour une mise en conformité.

Le poids de PCI-DSS risque de se renforcer rapidement, probablement par l'intermédiaire de réglementations au niveau européen. La traduction en loi française en découlera ensuite.

PS : Merci à Emmanuel Fleury pour la relecture.



Liens

[1] Site de Carte Bleue : **http://www.carte-bleue.com/upload/PDF1_29.pdf**

[2] Site de MasterCard : **http://www.mastercardfrance.com/fr/europay/societe/pdf/MasterCard_Europay_Chiffres_cles_2006.pdf**

[3] Site de la Banque de France : **http://www.banque-france.fr/observatoire/telechar/rap2006_chap2.pdf**

[4] Informations relatives à la fraude sur Internet : **http://www.journaldunet.com/0411/041102fraudesurinternet.shtml**

[5] Site Internet du PCI SSC : **https://www.pcisecuritystandards.org/index.htm**

Autres ressources :

http://www.pcieurope.com/

http://www.treasuryinstitute.org/blog/

http://en.wikipedia.org/wiki/PCI_DSS

Marion Videau – marion.videau@loria.fr – LORIA / Université Henri Poincaré, Nancy 1
Jean-Baptiste Bédrune – jb@security-labs.org – Ingénieur chercheur, SOGETI / ESEC

LES MODES DE CHIFFREMENT PAR BLOCS

mots clés : *chiffrement à clé secrète / primitive cryptographique / rétro-ingénierie*

1. Rappels sur le chiffrement

Un chiffrement est une transformation E d'un ensemble M de messages définis sur un alphabet A vers un ensemble de cryptogrammes C définis sur un alphabet A'. Afin que le destinataire d'un cryptogramme puisse le déchiffrer de manière unique, il faut que la transformation E soit inversible. On appelle déchiffrement la transformation inverse notée D, de l'ensemble C vers M.

La conception d'un chiffrement n'est pas une opération particulièrement connue pour être rapide ou facile – presque 4 ans ont été nécessaires entre l'appel à propositions et la standardisation, que ce soit pour le DES (1973-1977) ou l'AES (1997-2000). Ainsi, un des principes historiques de la conception de chiffrement énonce qu'il vaut mieux concevoir une famille de chiffrements dépendant d'un paramètre simple, appelé « clé », qu'il est possible de changer à sa guise [5]. La divulgation de la procédure de (dé-)chiffrement – considéré comme un événement devant se produire – ne doit pas mettre en danger le système dont la sécurité ne doit reposer que sur le secret de la clé. Lorsque nous parlerons d'algorithme de chiffrement, nous nous référerons implicitement à la famille de transformations plutôt qu'à une transformation donnée.

Un chiffrement est une transformation...

Un système de chiffrement repose ainsi sur trois ensembles :

- un ensemble de messages clairs M, sous-ensemble de A^* ;
- un ensemble de messages chiffrés C, sous-ensemble de A^* ;
- un ensemble de clés (chiffrement et déchiffrement) $K = (K_e, K_d)$,

ainsi que sur trois algorithmes :

- un algorithme de génération de clés ;
- un algorithme de chiffrement $\{E_e, e \in K_e\}$;
- un algorithme de déchiffrement $\{D_d, d \in K_d\}$.

Parmi les critères qui président aux choix d'un système, on trouve de manière évidente la sécurité, puisque l'utilisation des algorithmes ci-dessus implique forcément un surcoût. En conflit direct avec le précédent, on trouve la performance dès qu'il s'agit de l'utilisation en environnement réel. La question de la performance (au sens large) impacte tous les aspects et doit bien entendu prendre en compte la fréquence d'utilisation des divers algorithmes :

- génération de clés ;
- chiffrement ;
- déchiffrement,

ainsi que la distribution de clés. S'y ajoute la question du stockage des implémentations des algorithmes, et du stockage des instances pendant l'exécution : taille des exécutables ou des circuits, taille et conditions de conservation des clés.

Si l'on souhaite qu'il n'y ait pas d'expansion de la taille des messages – afin de ne pas faire chuter la bande passante, par exemple – on souhaite que pour tout message de longueur l, le chiffrement se comporte comme une permutation sur l'ensemble des mots de taille l, et ce, pour toute valeur de l. Comment concevoir un tel système ? L'idée naturelle qui apparaît lorsqu'on ne sait pas construire un objet complexe sur une entrée de grande taille consiste à découper l'entrée en blocs consécutifs de même taille et à concaténer des transformations sur ces entrées plus petites. La simplicité voudrait qu'on utilise la même transformation autant de fois que nécessaire. Malheureusement, cette idée est à exclure absolument, sauf cas particuliers, puisqu'elle ne permet pas de masquer la structure du texte clair sous-jacent. Illustrons le cas en prenant l'exemple immédiat où la taille de bloc est n bit. Nous enfoncerons à nouveau le clou un peu plus loin en illustrant le mode ECB.

Le cas $n = 1$ correspond au cas le plus répandu de chiffrement à flot. Dans ces systèmes, le chiffrement et le déchiffrement reposent sur une opération très simple, le OU-EXCLUSIF bit à bit qui définit une permutation pour chaque bit de texte clair, virtuellement pour toute longueur de message, la limite haute étant, par exemple 2^{80} bits. Si on concatène l fois la même permutation, on obtient soit le texte clair, soit son complémentaire. Ceci constitue évidemment l'exemple extrême, mais les cas où $n > 1$ se doivent en général d'obéir à la même règle. Dans le cas du chiffrement à flot, la permutation est très simple et on reporte la complexité du système sur l'algorithme de génération de clés. Dans le cas connu du *masque jetable*, il faut générer de l'aléa véritable, distribuer une clé aussi longue que le message qu'elle devra chiffrer, stocker de part et d'autre une telle clé et s'assurer de ne s'en servir qu'une unique fois. On comprend aisément qu'on puisse préférer générer une clé pseudo-aléatoire à partir d'une initialisation plus petite dont une partie servira de clé secrète, elle-même plus facile à partager. On perd en sécurité, mais on gagne en performance du côté de la génération de la *suite chiffrante*. On contourne également par ce biais une limitation pratique du masque jetable dont les caractéristiques le rendent peu utilisable – délicat euphémisme – pour chiffrer du stockage de données, un disque dur par exemple.

L'exemple précédent nous permet également d'illustrer une caractéristique des alphabets construits sur $\{0,1\}$: ils se réduisent tous à ce dernier. Ainsi, un message de M de taille l, multiple de n, peut indifféremment être considéré comme :

- une lettre de l'alphabet $A = \{0,...,2^l - 1\}$;
- un mot de l / n lettres de l'alphabet $A = \{0,...,2^n - 1\}$;
- un mot de n bits, lettres de l'alphabet $A = \{0,1\}$.

Cette caractéristique permet d'envisager les chiffrements de diverses manières. Dans l'exemple précédent du chiffrement à flot, on a considéré qu'on utilise une permutation de chaque bit paramétré par un bit pseudo-aléatoire. On peut également dire que la clé secrète K (à laquelle s'ajoute en général une valeur d'initialisation) paramètre une permutation sur toute chaîne binaire de l bits.

Comme nous l'avons souligné, on peut également travailler sur un alphabet de taille plus élevée, avec des lettres formées de n bits, soit que la structure du système l'impose (cas des systèmes de chiffrement à clés publiques actuellement utilisés), soit qu'on cherche à tirer parti de la structure en blocs de nombreuses données. Nous laisserons à partir d'ici totalement de côté les algorithmes à clé publique dont les utilisations les plus courantes concernent des messages de petite taille (clés secrètes dites « de session », hachés de messages) pour nous intéresser uniquement aux chiffrements par blocs à clé secrète qui requièrent des modes opératoires.

2. Le chiffrement itératif par blocs

Les chiffrements par blocs sont des primitives cryptographiques largement répandues. Ils peuvent en effet être vus comme des composantes cryptographiques élémentaires utilisables comme brique de base pour la construction d'autres primitives grâce à des combinaisons adaptées :

- fonction de compression à sens unique (par exemple, schéma de Meyer-Davies, Matyas-Meyer-Oseas ou Miyaguchi-Preneel) ;
- algorithme de code d'authentification de messages (par exemple, CBC-MAC) ;
- chiffrement à flot (par exemple, modes CFB, OFB ou CTR).

Les chiffrements par blocs sont des primitives cryptographiques largement répandues.

D'autre part, ils comptent parmi eux les standards de chiffrement que sont le DES désormais remplacé par l'AES. La très large utilisation de ces chiffrements a motivé leur étude et conduit en une quarantaine d'années à la formalisation, certes incomplète, de critères de conception stables s'appuyant sur un nombre réduit de schémas de base et l'ensemble des attaques publiées à ce jour.

Considérons un bloc de message m de taille n bits. Pour que le chiffrement E_K soit sûr, il faut qu'il soit impossible de retrouver m à partir de $c = E_K(m)$ sans aucune connaissance sur la clé K de taille k bits. Comme nous l'avons vu, E_K est une permutation sur l'ensemble des chaînes binaires de taille n. Il existe $2^n!$ telles permutations indexées par les 2^k valeurs possibles de la clé K. Concevoir un système de chiffrement par blocs E revient ainsi à faire le choix du sous-ensemble de permutations définies par la clé K. Idéalement, un tel ensemble de 2^k permutations ne devrait pas être distinguable d'un ensemble de 2^k permutations tirées aléatoirement parmi les $2^n!$ possibles.

Nous n'allons pas dans le cadre de cet article entrer dans les détails de conception de ces systèmes. Il est néanmoins important de noter qu'ils doivent assurer deux propriétés énoncées par Shannon [10] :

- la *confusion*. Cette propriété renvoie à l'explicitation mathématique des liens entre le clair, le chiffré et la clé : $c = E_K(m)$. Si, comme dans le cas du masque jetable, la transformation est un OU-EXCLUSIF bit à bit, alors, à partir d'un couple clair/chiffré, il est facile de retrouver K (d'où le caractère nécessairement *jetable* de la clé). Dans un chiffrement par blocs, la permutation E_K doit être suffisamment complexe pour que même à partir de nombreux couples clairs/chiffrés, il soit calculatoirement impossible de retrouver la clé K ;

⇨ la *diffusion*. Cette propriété renvoie aux caractéristiques statistiques que peuvent présenter les blocs de clair et la clé. En effet, un bloc de clair *m* est également une suite de *n* bits qui présentent souvent un profil statistique (lié au format, à la langue, etc.). Si ce profil statistique transparaît au travers du chiffré, c'est une faiblesse évidente du chiffrement.

La taille de la clé correspond à celle d'un chiffrement à clé secrète, à savoir la limite évaluée de la recherche exhaustive sur l'ensemble des chaînes binaires de taille *k*. Actuellement, la taille de clé est en général de 128 bits. Les tailles de blocs peuvent être de 64, 128 ou 256 bits, l'idée étant qu'on ne doit pas pouvoir décrire simplement le chiffrement par la correspondance pour chaque clé entre les clairs et les chiffrés (un dictionnaire en quelque sorte).

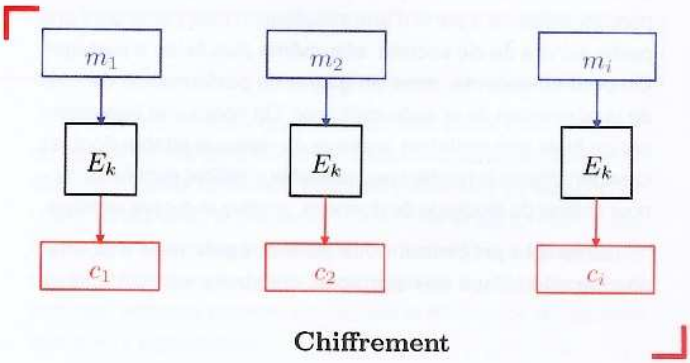
Les chiffrements à clé secrète étant utilisés pour chiffrer des masses importantes de données du fait de leur rapidité, il est fort peu probable que leur utilisation se limite à des données de taille inférieure à la taille de blocs. Comment chiffrer des données de taille supérieure à la taille de bloc ? C'est l'objet des modes opératoires.

3. Les modes opératoires

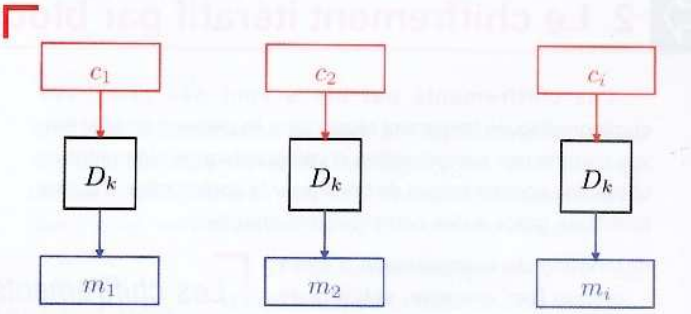
Les cinq modes principaux (ECB, CBC, CFB, OFB et CTR) sont décrits par le NIST, entre autres, sur la page *Block cipher modes* à l'adresse **http://csrc.nist.gov/groups/ST/toolkit/BCM/index.html** Les quatre premiers ont été originellement spécifiés pour le DES et décrits dans le standard FIPS 81 [7] pour une taille de blocs de 64 bits. Le mode CTR a été rajouté dans la publication spéciale du NIST, NIST SP 800-38A [8]. Ces modes ont également été repris par la norme ISO/IEC 10116 [4] pour une taille de bloc de *n* bits. Les modes ECB et CBC sont qualifiés de modes par blocs alors que les modes CFB, OFB et CTR sont qualifiés de modes par flots. Il existe par ailleurs d'autres modes opératoires, qui permettent notamment de garantir simultanément la confidentialité et l'intégrité des données (on parle de *chiffrement authentifié*).

Les schémas des modes opératoires nous ont été fournis par Cédric Lauradoux [1], qui les a placés pour l'occasion sous Copyleft.

Les blocs de clair sont notés $m_1,\ldots,m_i$ et les blocs de chiffré correspondant $c_1,\ldots,c_i$.



Chiffrement



Dechiffrement

3.1 ECB, Electronic Code Book

Pourquoi parler du mode à ne pas utiliser en pratique ? Tout simplement, car il est une bonne illustration de l'affrontement entre sécurité et performances. En effet, si ce mode est cryptographiquement faible du fait de l'indépendance des blocs (on peut ainsi en modifier l'ordre, en insérer de nouveaux, reconnaître la structure du message sous-jacent), cette même indépendance permet de paralléliser le chiffrement et le déchiffrement et d'avoir accès à un bloc précis sans avoir à chiffrer ou déchiffrer tout le message.

La caractéristique (bonne et mauvaise) de ce mode provient du fait que la clé est identique à chaque transformation et que le message conserve sa structure par découpage simple. Ainsi, il n'est pas recommandé de l'utiliser pour le chiffrement de plus d'un bloc de message ou si la clé doit être utilisée pour plus d'un bloc de message.

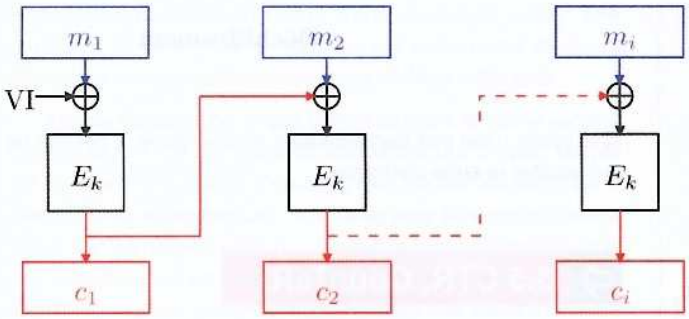
1 http://www.princeton.edu/~claurado/index.html, remerciements à lui qui nous a sauvé de quelques heures de rédaction.

2 Nous utiliserons donc pour les décrire la modélisation d'un automate à états finis constitué d'un état interne, d'une fonction de transition d'état et d'une fonction de filtrage qui fournit les bits de pseudo-aléa. Nous avons décrit ce modèle dans le numéro précédent de MISC.

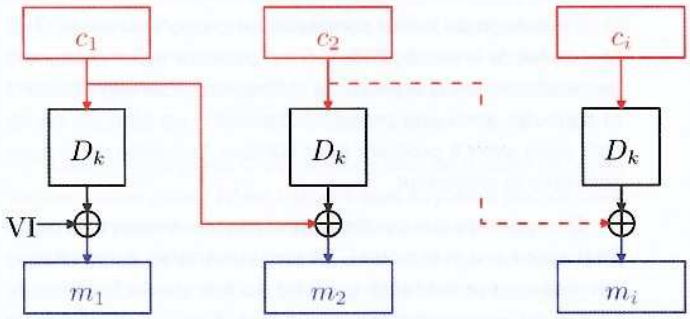
En s'inspirant de l'exemple du chiffrement à flot, on pourrait être tenté de modifier la clé à chaque application de la fonction de chiffrement en produisant une suite pseudo-aléatoire. Cette option ne trouve guère d'applications, sans aucun doute par le surcoût élevé à payer en termes de performances qu'elle impliquerait. L'option qui semble a priori la plus simple et la moins coûteuse à mettre en œuvre consiste à masquer de manière simple la structure du bloc de message avant de procéder à son chiffrement : c'est le mode CBC. Afin d'améliorer les performances et la robustesse aux erreurs de transmission, on peut faire appel aux modes CFB, OFB et CTR qui transforment un chiffrement par blocs en chiffrement à flot [2].

3.2 CBC, Cipher Block Chaining

Dans le mode CBC, on ajoute à chaque bloc de clair le chiffré du bloc précédent par un OU-EXCLUSIF bit par bit avant de le soumettre au chiffrement par blocs. Le premier bloc de clair (qui n'a donc pas de bloc de chiffré précédent) est ajouté à une valeur initiale, notée *IV*, valeur qui est en général publique et qui varie à chaque nouveau chiffrement. Cette valeur garantit que le chiffrement de deux messages identiques produira deux chiffrés différents.



Chiffrement



Déchiffrement

Ce mécanisme garantit que chaque bloc de chiffré dépend de tous les blocs de clair qui le précèdent. Cette caractéristique est très utile pour construire un mode authentifiant. Malheureusement, cette même caractéristique empêche tout accès direct à un bloc en lecture/écriture sans passer par le chiffrement ou le déchiffrement de tous les blocs précédents ce qui en fait un mode peu adapté à du stockage de masse.

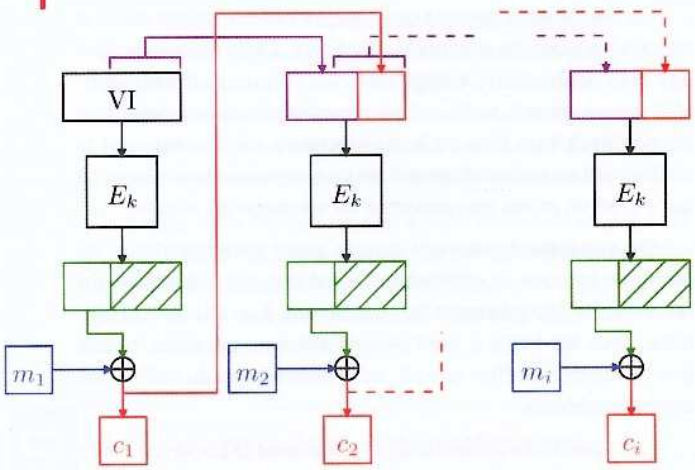
On constate également qu'une autre caractéristique de ce mode est que le chiffrement n'est pas parallélisable. En revanche, le déchiffrement l'est quasiment, à un OU-EXCLUSIF près : tous les blocs c_i sont déchiffrables en parallèle, tandis que les blocs à chiffrer avec E_K sont dépendants du chiffré des blocs précédents.

Les opérations nécessitent, comme avec ECB, la présence d'un bloc complet avant de pouvoir débuter. Par ailleurs, une erreur d'un bit pendant la transmission du chiffré affecte tout le bloc de clair correspondant obtenu après déchiffrement. Enfin, si la longueur du message n'est pas un multiple de la taille de blocs, l'ajout de bits de remplissage est nécessaire sauf à utiliser la méthode dite « *de vol de chiffré* » [2].

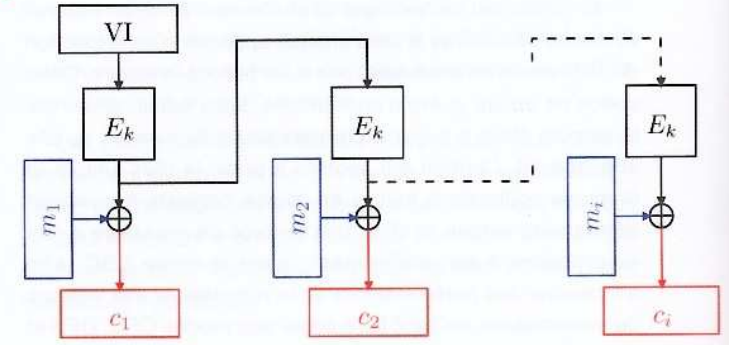
3.3 CFB, Cipher FeedBack

Le mode CFB améliore le problème de la taille de bloc. On conçoit pour ce faire un chiffrement à flot auto-synchronisant. C'est un chiffrement à flot pour lequel chaque bloc à chiffrer est dépendant des blocs chiffrés précédents. L'état interne est initialisé par un vecteur d'initialisation *IV*. Le rôle de la fonction de filtrage est tenu par le chiffrement par blocs auquel on associe la fonction de troncature à $r \leq n$ bits. La fonction de transition, quant à elle, consiste à décaler l'état interne et à y adjoindre les r bits de chiffrés obtenus par le OU-EXCLUSIF bit à bit de r bits du message avec ceux de la fonction de filtrage.

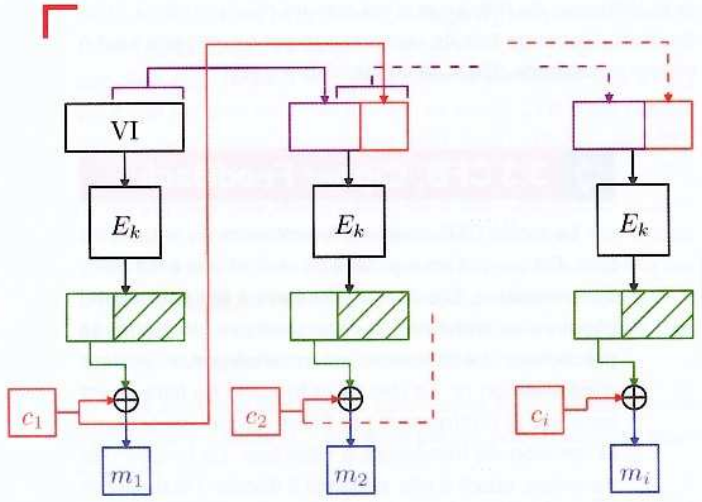
De manière assez similaire au mode CBC, le chiffrement n'est pas parallélisable et le déchiffrement l'est quasiment. Ce dernier, contrairement aux modes ECB et CBC ne nécessite que la fonction de chiffrement par blocs (et non celle de déchiffrement). Cela a l'avantage d'économiser un programme. Les opérations nécessitent la présence d'un bloc de r bits avant de pouvoir débuter, r étant plus petit, cela nécessite moins de mémoire tampon. Par ailleurs, une erreur d'un bit pendant la transmission du chiffré affecte tout le bloc de clair suivant après déchiffrement. Enfin, si la longueur du message n'est pas un multiple de la taille de blocs, l'ajout de bits de remplissage n'est pas nécessaire : il suffit de jeter les bits de suite chiffrante surnuméraires.



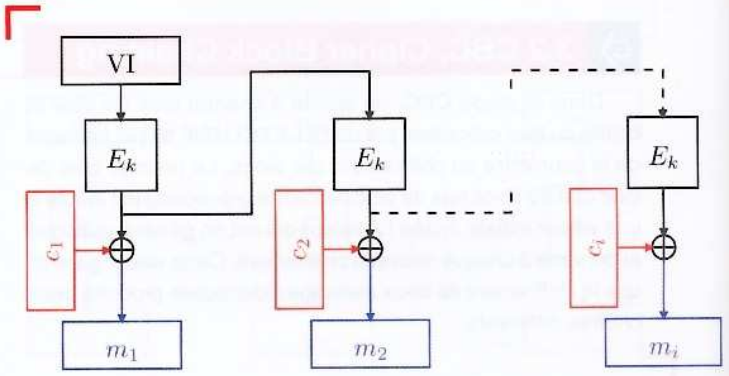
Chiffrement



Chiffrement



Déchiffrement



Déchiffrement

3.4 OFB, Ouput FeedBack

Le mode OFB est un chiffrement à flot synchrone. Ce type de chiffrement est une combinaison, généralement un OU EXCLUSIF, entre le message à chiffrer et une suite binaire de même longueur. Cette dernière est indépendante, et du message clair, et du message chiffré.

L'état interne est initialisé par un vecteur d'initialisation *IV*. Le rôle de la fonction de filtrage est tenu par le chiffrement par blocs (qui est également la fonction de transition) auquel on associe la fonction de troncature à $r \leq n$ bits. On utilise en général le mode OFB avec $r = n$.

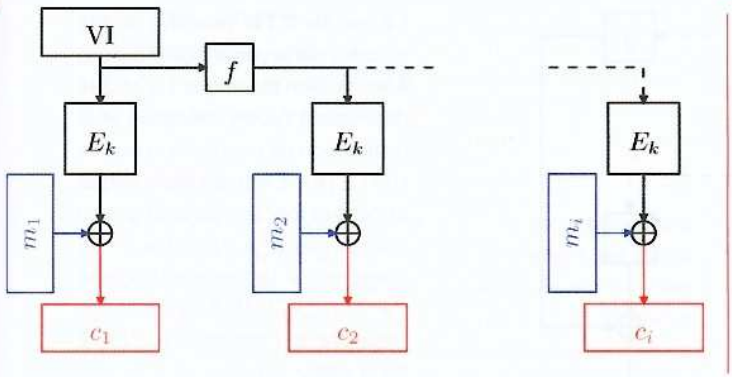
Tout comme les chiffrements à flot synchrone de manière générale, ce mode ne propage pas les erreurs et les opérations de chiffrement et de déchiffrement sont identiques. Par ailleurs, si le mode n'est pas parallélisable en lui-même, il permet de précalculer la suite chiffrante.

3.5 CTR, CounTeR

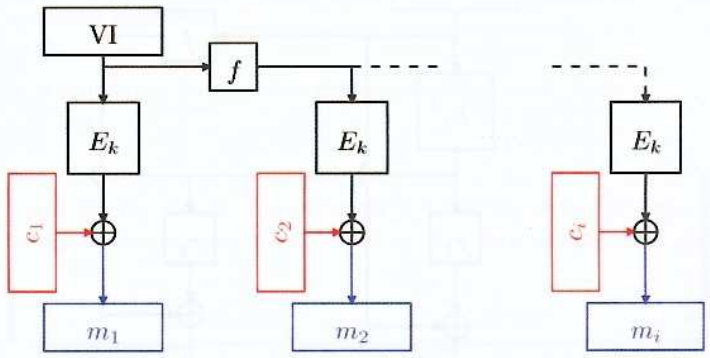
Le mode compteur, comme le mode OFB, est un chiffrement à flot synchrone. L'état interne est initialisé par un vecteur d'initialisation *IV*. Le rôle de la fonction de filtrage est tenu par le chiffrement par blocs. La fonction de transition est une fonction simple qui vaut à l'état interne le nom de compteur.

L'avantage du mode compteur par rapport au mode OFB tient au fait de la simplicité de la fonction de transition qui permet de paralléliser les opérations de chiffrement et de déchiffrement et d'ajouter ainsi une propriété d'accès à un bloc donné du clair sans avoir à produire pour l'obtenir la totalité de la suite chiffrante le précédant.

D'un point de vue de générateur pseudo-aléatoire, le mode OFB avec $r = n$ et le mode CTR sont vulnérables à une attaque par distingueur évidente qui tient au fait que la fonction de filtrage est une permutation sur un bloc. Ainsi, il n'existe pas de collisions entre blocs de taille n dans la suite chiffrante produite, ce qui devrait être le cas après $2^{n/2}$ blocs produits pour une suite réellement aléatoire, d'après le paradoxe des anniversaires.



Chiffrement



Déchiffrement

4. Problème lié à la valeur initiale

4.1 La nécessité d'une valeur initiale

Comme nous l'avons vu, les quatre derniers modes opératoires que nous avons décrits ci-dessus nécessitent tous un vecteur d'initialisation, tout comme les chiffrements à flot. Cette chaîne binaire est généralement publique, évitant le problème du maintien secret d'une deuxième valeur. Elle permet d'assurer que le chiffrement consécutif de deux messages identiques fournisse deux chiffrés différents.

Afin de s'assurer de ne pas utiliser toujours le même vecteur d'initialisation, ce qui ruinerait son utilité, un moyen simple est de prendre la valeur d'un compteur qui est incrémentée à chaque envoi et de longueur suffisante pour éviter un bouclage rapide sur les mêmes valeurs. Lorsque le vecteur d'initialisation doit en outre avoir un caractère (pseudo-)aléatoire, il peut être déduit d'un nonce auquel on applique un tour du chiffrement par blocs ou issu d'un générateur pseudo-aléatoire.

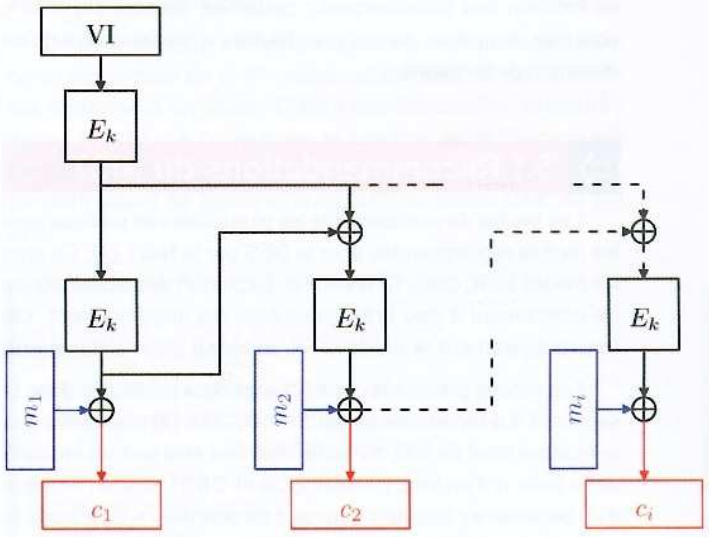
L'injection du vecteur d'initialisation a longtemps constitué le talon d'Achille des générateurs pseudo-aléatoires cryptographiques et les modes opératoires concernés n'y échappent bien évidemment pas. La formalisation du problème de l'injection d'*IV* pour un chiffrement à flot est décrite dans [1]. Si on considère par exemple le mode OFB, avec $r = n$, ou le mode CTR, tels que décrits ci-dessus, ils sont tous les deux vulnérables à une attaque par distingueur avec valeur initiale choisie.

Ainsi, pour le mode CTR, le *i*-ème bloc de la suite générée par une valeur initiale *IV* est toujours égal au *j*-ième bloc de celle générée à partir de la valeur initiale $f^{(i-j)}(IV)$. De la même façon, pour le mode OFB, le *i*-ème bloc de la suite générée par une valeur initiale *IV* est toujours égal au *j*-ième bloc de celle générée à partir d'une valeur initiale égale à la sortie du *(i-j)*-ème tour. Il est donc nécessaire de s'assurer qu'aucun bloc de suite chiffrante produit par OFB avec une clé *K* ne soit utilisé comme *IV* pour ce mode avec la même clé pour chiffrer des messages ultérieurs.

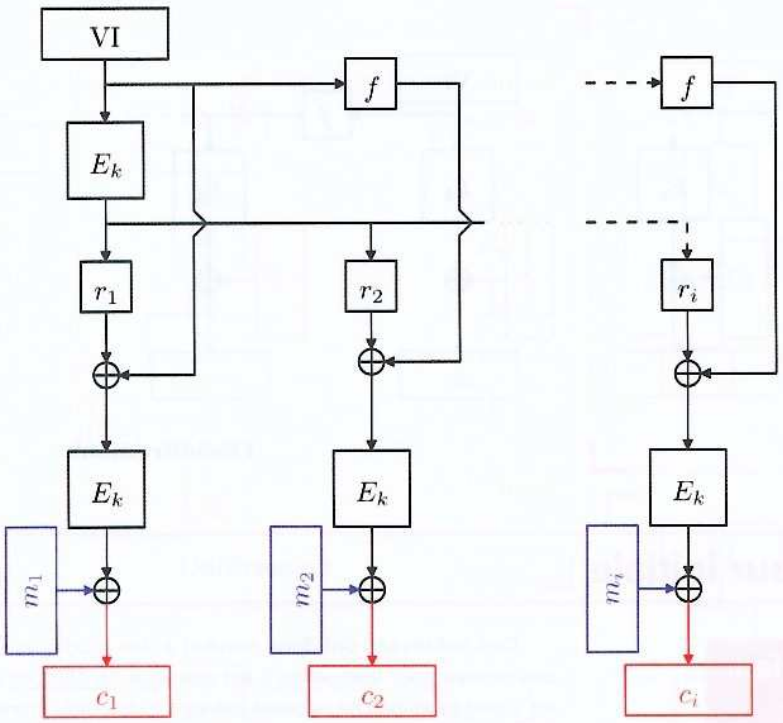
Ces faiblesses ont donc conduit à des modifications de ces modes pour lesquelles il est possible de démontrer que les suites produites ne peuvent pas être distinguées d'une suite aléatoire si le chiffrement par blocs sous-jacent possède une propriété similaire.

4.2 Les modes OFB et CTR modifiés

Le mode OFB modifié a été introduit en 1985 par la norme américaine ANSI X9.17 (*Financial Institution Key Management (Wholesale) standard*) pour la construction de générateurs pseudo-aléatoires pour le chiffrement à flot. Afin de limiter les possibilités d'attaques à IV choisies, l'idée consiste à ajouter $E_K(IV)$ à chaque entrée avant l'appel à la fonction de chiffrement.



Chiffrement



Chiffrement

Le mode CTR modifié a été introduit par le projet 3GPP dans la spécification technique TS 35.206 définissant l'authentification et la distribution de clés dans la norme UMTS [6]. L'entrée de chaque appel à la fonction de chiffrement correspond à la valeur du compteur à laquelle est ajouté par OU-EXCLUSIF le décalage circulaire de r_i positions du chiffré de la valeur initiale.

Des résultats théoriques sur le caractère pseudo-aléatoire des suites produites par ces deux modes modifiés ont été publiés dans [3].

5. Les mises en œuvre en pratique

De nombreuses bibliothèques de chiffrement de qualité variable sont disponibles. Il est souvent conseillé de réutiliser des bibliothèques en cryptographie, plutôt que de réinventer la roue : l'implémentation des algorithmes de chiffrement est souvent pénible, et les opérations critiques, comme la génération de clés ou d'aléas, sont difficiles à implémenter correctement.

Les fonctions de chiffrement par blocs implémentées varient en fonction des bibliothèques : certaines, comme `Crypto++`, sont très complètes, tandis que d'autres n'implémentent qu'un minimum de fonctions.

5.1 Recommandations du NIST

Les modes de confidentialité les plus utilisés en pratique sont les modes recommandés pour le DES par le NIST [7]. Ce sont les modes ECB, CBC, CFB et OFB. La plupart des bibliothèques de chiffrement à peu près complètes les implémentent. On trouve également le mode CTR, autorisé plus tardivement.

Les modes présentés dans [7] sont ceux présentés dans la section 3. La recommandation SP 800-38A [8] préconise que le nombre total de bits du texte clair doit être soit un multiple de la taille n d'un bloc (modes ECB et CBC), soit un multiple d'un paramètre r appelé « segment de données », inférieur à la taille d'un bloc (CFB). Les modes OFB et CTR n'imposent pas de conditions sur la taille du message.

En pratique, on choisit $r = 8$ (8 bits = 1 octet, facile à manipuler) ou $r = n$ pour le mode CFB. On ajoute en général cette dernière valeur à la fin du nom du mode de chiffrement (CFB128 pour AES, par exemple).

L'accréditation FIPS 140-2 est possible pour ces modes. D'autres modes de confidentialité plus exotiques sont parfois ajoutés dans des bibliothèques. Ceux-ci ne peuvent pas être validés.

5.2 Quelques implémentations

Cette partie recense les modes de chiffrement implémentés dans des bibliothèques fréquemment utilisées, et détaille la manière dont ils sont implémentés.

Certaines bibliothèques respectent les conditions nécessaires de FIPS 140-2. Elles sont généralement disponibles en deux versions : une version certifiée, contenant un nombre restreint de fonctions de chiffrement et de modes opératoires, et une autre version plus étoffée.

Miracl

`Miracl` (*Multiprecision Integer and Rational Arithmetic C/C++ Library*) est, comme son nom l'indique, une bibliothèque de gestion des grands nombres. Elle est orientée cryptographie

asymétrique. Afin de la compléter, un algorithme de chiffrement symétrique (AES) a été implémenté. En plus des modes de FIPS 81, elle gère le mode PCFB (*Propagating Cipher Feed-Back*) créé par les développeurs [9].

OpenSSL

La bibliothèque de chiffrement d'OpenSSL est très complète en ce qui concerne les algorithmes de chiffrement par blocs. C'est une référence, bien que son utilisation ne soit pas des plus simples. Il existe une version FIPS 140-2 sortie en février 2007, beaucoup plus réduite. Une nouvelle version est en cours de validation.

Tous les modes de FIPS 81 sont implémentés pour chaque algorithme. Le mode CTR est disponible pour AES et Camellia.

Crypto++

`Crypto++` est une bibliothèque fortement orientée objet, très complète, écrite en C++. Tous les modes définis précédemment sont implémentés. Le mode CFB est utilisable pour toute valeur de taille inférieure à un bloc de chiffrement. Une version validée FIPS 140 est disponible, avec les restrictions décrites plus haut sur le mode CFB.

DCPCrypt

`DCPCrypt` est la principale bibliothèque de chiffrement utilisée par les programmeurs Delphi, et est très simple d'utilisation. Elle intègre tous les modes recommandés par le NIST. Le mode CBC est modifié pour pouvoir travailler sur des données de longueur non congruente à la taille d'un bloc de chiffrement : les données de fin sont chiffrées en mode OFB.

5.3 Méthodes d'implémentation

On peut distinguer deux méthodes principales pour l'implémentation des modes de chiffrement : une approche impérative, où le mode de chiffrement et la fonction de chiffrement sont fortement liées ; et une approche objet, où la fonction de chiffrement et le mode utilisés sont deux objets différents.

Méthode impérative

Dans le premier cas, on implémente les fonctions de chiffrement et de déchiffrement d'un bloc, et on écrit ensuite une fonction différente pour chaque mode de chiffrement.

C'est par exemple l'approche de `Miracl`, qui ne comporte qu'une fonction de chiffrement, et, plus étonnamment, d'OpenSSL. Les fonctions associées au chiffrement AES avec OpenSSL sont les suivantes :

- chiffrement d'un bloc : `AES_encrypt` et `AES_decrypt` ;
- modes associés : `AES_ecb_encrypt`, `AES_cbc_encrypt`, etc.

Le prototype de la fonction chiffrant des données en mode CBC est :

```
void AES_cbc_encrypt(const unsigned char *in, unsigned char *out,
  const unsigned long length, const AES_KEY *key, unsigned char
  *ivec,
  const int enc);
```

Pour `Miracl`, le mode de chiffrement est passé en argument à la fonction d'initialisation, qui le stocke dans un contexte. La fonction de chiffrement `aes_encrypt` comporte un `switch` sur le mode utilisé, et chiffre les données en conséquence.

Dans ces situations, il faut écrire pour chaque fonction de chiffrement autant de fonctions qu'il y a de modes associés. Le nombre de fonctions à écrire dans une bibliothèque complète devient vite très gros. Cette approche a pour avantage une plus grande rapidité d'exécution.

Approche objet

Avec une approche objet, telle celle employée dans `DCPCrypt`, on implémente une classe générique `TDCP_blockcipher`, ayant pour méthodes membres entre autres les fonctions de chiffrement associées à chaque mode : `EncryptCBC`, `EncryptOFB`, etc. Le mode étant indépendant de la fonction de chiffrement, ces méthodes sont donc génériques, à la taille du bloc de chiffrement près : en fait, deux classes génériques sont dérivées de `TDCP_blockcipher`, une pour les chiffrements par blocs de 64 bits et l'autre pour ceux de 128 bits. Seules les méthodes `EncryptECB` et `DecryptECB`, correspondant aux fonctions de chiffrement et déchiffrement d'un bloc ont alors besoin d'être surchargées pour chaque algorithme de chiffrement.

On peut aller plus loin dans l'approche objet en considérant qu'un algorithme de chiffrement est un objet, et qu'un mode est également un objet. C'est l'implémentation effectuée dans `Crypto++`. À l'utilisation, la fonction de chiffrement est un objet passé en paramètre du constructeur de l'objet mode. Le chiffrement de données avec AES en mode CBC se fait ainsi :

```
byte key[16], iv[16], plaintext[100];
std::string ciphertext;
AESEncryption aesEncryption(key, 16);
CBCPaddedEncryptor cbcEncryptor(aesEncryption, iv, new
                  StringSink(ciphertext));
cbcEncryptor.Put(plaintext, 100);
cbcEncryptor.MessageEnd();
return ciphertext;
```

6. Rétro-conception

L'étude de logiciels contenant des procédés cryptographiques est souvent utile pour vérifier la qualité d'une solution ou vérifier sa résistance. Si le code du logiciel n'est pas disponible, un travail de *reverse engineering* est parfois indispensable.

Il est bien souvent très facile de déterminer les fonctions de chiffrement utilisées, notamment en recherchant les *S-boxes* référencées dans le binaire. Lorsqu'on ne possède pas les signatures des bibliothèques utilisées, déterminer le mode utilisé est une autre histoire, qui peut s'avérer (vraiment) coûteuse en temps.

Une fonction chiffrant des données va toujours appeler E_K ou D_K, faciles à identifier. On trouve les fonctions correspondant aux modes en regardant les références à E_K et D_K. Si les implémentations des bibliothèques sont différentes, les algorithmes restent les mêmes. Et ils sont tous basiques.

On pourrait regarder le nombre d'arguments passés à chacune de ces fonctions pour deviner le mode. Ou encore analyser dynamiquement le programme en injectant à ces fonctions des vecteurs de test. Une approche décomposant chacune des fonctions en éléments simples donne des résultats beaucoup plus rapides.

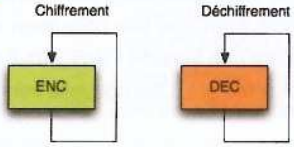
Les algorithmes correspondant aux modes de chiffrement ne nécessitent que quelques opérations :

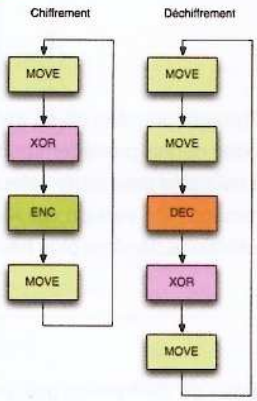
- une copie de bloc `COPY` ;
- un OU EXCLUSIF sur place entre deux blocs `XOR` ;
- un chiffrement `ENC` ou un déchiffrement `DEC` ;
- l'incrémentation d'un compteur `INC` pour le mode CTR.

La structure des fonctions associées aux modes sera toujours une combinaison de ces 4 opérations. Les schémas suivants, découlant des graphes de la section 3, présentent ces structures.

6.1 ECB

C'est le mode le plus simple. Seuls `ENC` ou `DEC` sont appelés.



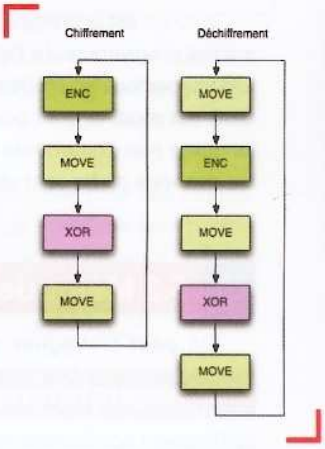


6.2 CBC

CBC est le seul mode, après ECB, qui utilise D_K dans la routine de déchiffrement. Il est facile à repérer quand les fonctions de chiffrement et de déchiffrement sont présentes dans le programme.

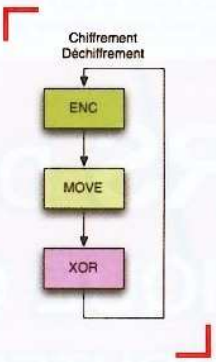
6.3 CFB

Dans les trois derniers modes, D_K a disparu. La décomposition en blocs basiques est très utile pour ces modes. Ils ont presque la même structure, mais, en gardant ces schémas sous les yeux, il ne devrait plus y avoir de problème pour les distinguer.



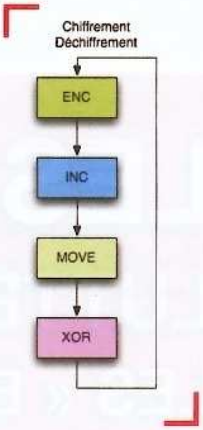
6.4 OFB

Contrairement à CFB, la structure des fonctions de chiffrement et de déchiffrement est identique. Il s'agit en fait de la même fonction, le mode correspondant à un flot synchrone.



6.5 CTR

Le chiffrement et le déchiffrement sont identiques encore une fois. L'incrémentation du compteur, qui parfois n'est pas *inliné* dans les programmes, rend la détection de CTR plus facile.



Conclusion

Il est indispensable d'associer un mode de chiffrement à un algorithme de chiffrement symétrique lorsqu'on souhaite chiffrer un message. Les approches les plus intuitives, comme souvent en cryptographie, sont à proscrire : le mode ECB, qui paraît couler de source, n'est à utiliser qu'avec beaucoup de précautions, par exemple.

Les algorithmes des modes présentés sont très simples. Il peut paraître alors assez trivial de concevoir de nouveaux modes. Néanmoins, cette facilité n'est qu'une illusion, la moindre modification d'un mode pouvant entraîner des vulnérabilités importantes et compromettre la sécurité des données à protéger.

Bibliographie

[1] BERBAIN (C.) et GILBERT (H.), « *On the security of IV dependant stream ciphers* », *Fast Software Encryption – FSE 2007, 4593 Lecture Notes in Computer Science*, 254-273, Springer-Verlag, 2007.

[2] DAEMEN (J.), *Cipher and Hash Function design*, Thèse de doctorat, Katholieke Universiteit Leuven, 1995.

[3] GILBERT (H.), « *The security of One-Block-to-Many Modes of Operation* », *Fast Software Encryption – FSE 2003, 2887 Lecture Notes in Computer Science*, 376-395, Springer-Verlag, 2003.

[4] ISO/IEC 10116, « *Information technology – Security techniques – Modes of operation for an n-bit block cipher* », *International Organization for Standardization*, 1997.

[5] KERCKHOFFS (A.), « La cryptographie militaire », *Journal des sciences militaires*, vol. IX, pp. 5-38, janvier 1883, pp. 161-191, février 1883.

[6] 3rd Generation PartnerShip Project, « *3GPP TS 35.909 – Specification of the MILENAGE algorithm set: An example algorithm Set for the 3GPP Authentication and Key Generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 5: Summary and results of design and evaluation* », 2004. **http://www.3gpp.org/ftp/Specs/html-info/35909.htm**

[7] NIST, *DES Modes of Operation*, December 2, 1980. FIPS Publication 81.

[8] NIST, *Recommendation for block cipher modes of operation*, 2001. NIST Special Publication 800-38A.

[9] SCOTT (M.) et SHAFA'AMRY (M.), *Novel Chaining Methods for Block Ciphers*, Dublin City University, *School of Computer Applications Working Paper CA-1993*.

[10] SHANNON (C.), « *Communication theory of secrecy systems* », *Bell Systems Technical Journal*, 28(4), 656-715 (1949).

Philippe Evrard et Éric Filiol – Laboratoire de virologie et de cryptologie, ESAT
efiliol@esat.terre.defense.gouv.fr – philippe.evrard@esat.terre.defense.gouv.fr

LES ACTEURS DE LA LUTTE INFORMATIQUE OFFENSIVE : LES « BONS », LES « BRUTES » ET LES « TRUANDS »...



De nombreux acteurs sont partie prenante dans la lutte informatique offensive (LIO) sous quelque forme qu'elle soit envisagée. Certains sont fortement contraints par des aspects légaux de l'emploi de l'arme informatique qu'ils s'astreignent à respecter...

mots clés : *LIO / acteurs / attaques / terrorisme*

D'autres, poursuivant des buts idéologiques ou « d'intérêt » s'affranchissent de ce point de vue légal. Les derniers, enfin, ne poursuivent qu'un but lucratif et y emploient toutes les ressources à leur disposition. Mais quelle que soit la motivation guidant ces attaques, elles posent plusieurs interrogations et en particulier sociétales : quel est le niveau réel de protection face à ces « menaces », toute protection est-elle illusoire et quelles « menaces » font elles peser sur les libertés et les valeurs démocratiques. En effet, quand un « bon » passe du côté « brute », mais avec des moyens et un savoir-faire très sophistiqués, plus aucune donnée n'est potentiellement à l'abri. Ce sont précisément ces craintes sociétales qui ont pendant longtemps interdit toute réflexion sur la lutte informatique offensive. Le potentiel offert par la LIO, et en premier lieu par les techniques virales sophistiquées, permet de contourner absolument toutes les protections envisageables, en particulier lorsqu'elles sont déployées préventivement ou en amont. Mais, la peur « primale » du citoyen face à ces risques doit être tempérée par la raison. D'une part, la sécurité de chacun d'entre nous passe obligatoirement par un sacrifice même minimal de nos « chères libertés », car, au fond, on ne peut pas réclamer ces libertés et dans le même temps les refuser à tout acteur malfaisant potentiel. D'autre part, il faut mettre fin à la vision stérile et stupide qui consiste à penser que l'Etat a du temps à perdre à espionner chacun d'entre nous. L'État a déjà fort à faire avec les vrais méchants pour ne pas gaspiller du temps et des ressources précieuses à des futilités. Au final, la meilleure protection contre toute déviance – au fond, pourquoi la règle du pas vu pas pris se limiterait elle aux excès de vitesse – reste la défense de nos valeurs démocratiques. Et là, c'est la responsabilité de chacun d'entre nous et c'est une part essentielle dans la sécurité des systèmes de notre pays. La boucle est ainsi bouclée.

Le potentiel offert par la LIO, et en premier lieu par les techniques virales sophistiquées, permet de contourner absolument toutes les protections envisageables...

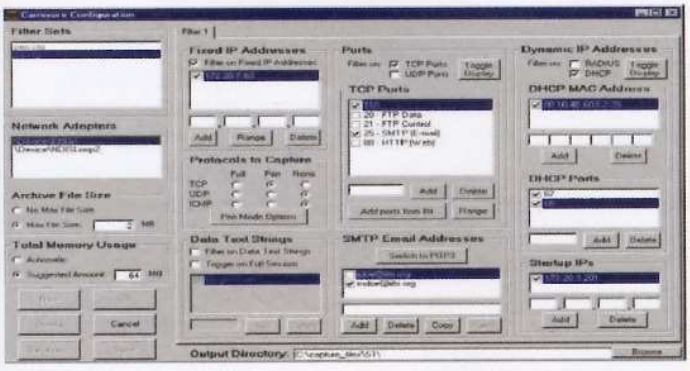
Les « bons »...

Lutte contre la criminalité

Dans ce cadre, l'informatique est de plus en plus utilisée par les « méchants ». La réaction, tout à fait normale et légitime, conduit à un début d'emploi de l'informatique « offensive » dans le cadre de la lutte contre la criminalité. Cet emploi a été mis à l'essai, avec plus ou moins de bonheur, dans plusieurs pays. On semble maintenant se diriger vers un emploi assez généralisé, mais dans un cadre précis et très encadré par les instances juridiques afin d'éviter les dérapages éventuels. Les précurseurs en la matière ont été les États-Unis.

Les États-Unis ont utilisé et utilisent depuis plusieurs années de manière active l'outil informatique dans la collecte du renseignement afin de lutter contre les organisations mafieuses et/ou terroristes. Dès 1999, le FBI a utilisé des outils de type « enregistreur de frappe » (*keylogger*). Un exemple célèbre, datant de 1998, a été le tristement célèbre logiciel DIRT, d'origine sud-africaine, ensuite racheté par la société américaine Codex Data Systems. Ce logiciel de type cheval de Troie s'installait lorsque la personne consultait certaines pages HTML ou bien via des documents de type Word ou Excel [1]. La partie module serveur ensuite enregistre et envoie vers le « centre de contrôle » toutes les données tapées au clavier.

Le plus célèbre de leurs outils a sans doute été le ver « *Magic Lantern* », utilisé dans le cadre du projet *Carnivore* [2] (Figure 1). Celui-ci a été la source d'un énorme scandale à la fin de 2001, lorsque le *Washington Post* a révélé que le FBI utilisait, pratiquement sous son seul contrôle, un ver permettant de faire sortir de l'information d'ordinateurs cibles. Ce scandale, à l'époque, était amplifié par la suspicion d'entente avec le FBI de certains éditeurs d'antivirus (voir infra : les éditeurs d'antivirus dans tout ça ?).



L'interface de configuration de Carnivore donne un très bon aperçu de ses possibilités. Cette interface n'a rien à envier à celle des chevaux de Troie modernes. Selon l'enquête qui a suivi la révélation du projet Carnivore au grand public, Magic Lantern n'aurait jamais été utilisé de manière opérationnelle.

Depuis, le FBI, toujours dans le même cadre, utilise un logiciel plutôt assimilé à un *spyware*. D'un emploi très encadré par le Département de la Justice (DoJ – *Department of Justice*) d'une catégorie de logiciels baptisée *Computer Internet Protocol Adress Verifier* (CIPAV – voir encadré 1, page suivante) [3].

Les autres pays aussi

L'informatique est maintenant une technologie communément répandue, utilisée universellement. Il est logique et normal que ces techniques évoluant et son emploi se généralisant, les nations envisagent de l'utiliser dans le cadre d'enquêtes de police. Cet emploi est étudié dans de nombreux autres pays : Allemagne, Autriche, Suisse, France [4]... De même qu'aux États-Unis actuellement, il s'agit là d'un emploi qui ne sera autorisé que dans certains cas très précis (en France, ont été évoqués : la pédophilie, le terrorisme, la criminalité en bande organisée) et, d'après ce que l'on peut en juger des débats auxquels ces études donnent lieu, strictement encadré par les ministères de la Justice. Il ne s'agit là, au final, que d'une simple évolution technologique qui est dans l'ordre des choses, de la même manière que, dans le cadre de certaines enquêtes, l'emploi d'écoutes téléphoniques ne choque personne.

Les éditeurs d'antivirus dans tout ça ?

La révélation de l'emploi de CIPAV par la presse a fait resurgir le spectre des dérapages de « Magic Lantern », conduisant à s'interroger sur la position des éditeurs de produits de sécurité face à ce genre de logiciels [5]. À cette occasion, ZDNet a publié une enquête réalisée auprès des grands éditeurs de sécurité [6].

Bilan : les réactions vont de « nous n'obéirons jamais à une telle requête » à « nous ne commentons pas les contacts que nous avons avec le département de la Justice » en passant par « la question suivante... » !

Aspect militaire [7]

D'autres acteurs du « monde de l'information » s'intéressent à l'emploi offensif de l'informatique tout en s'inscrivant dans des règles d'emploi et un cadre légal qui leur imposent de fortes contraintes. Il s'agira là presque essentiellement d'organisations internationales ou de nations qui, de par l'action entreprise, se voient contraintes à respecter ces règles. La multinationalité dans ce domaine impose généralement qu'un consensus se dégage. On en arrive alors, pour l'essentiel, à considérer l'aspect militaire de l'utilisation de l'informatique.

encadré 1

Un exemple de LIO ? CIPAV, le spyware du FBI – l'enquête « Timberlinebombinfo »

En juin 2007, le FBI a utilisé, dans le cadre d'une enquête sur des menaces d'attentat à la bombe dans un collège aux USA, un logiciel de type spyware. Cette utilisation de logiciel espion dans le cadre d'une enquête est, après le scandale « Carnivore/Magic Lantern » de 2001, strictement encadrée par le DoJ.

La déclaration sous serment de l'agent spécial du FBI demandant l'autorisation d'utiliser ce logiciel dans le cadre de l'enquête en question, et le mandat du DoJ sont accessibles sur Internet. Ce document de 18 pages donne des informations sur le cadre d'emploi de cet outil dans le déroulement de l'enquête.

Le CIPAV

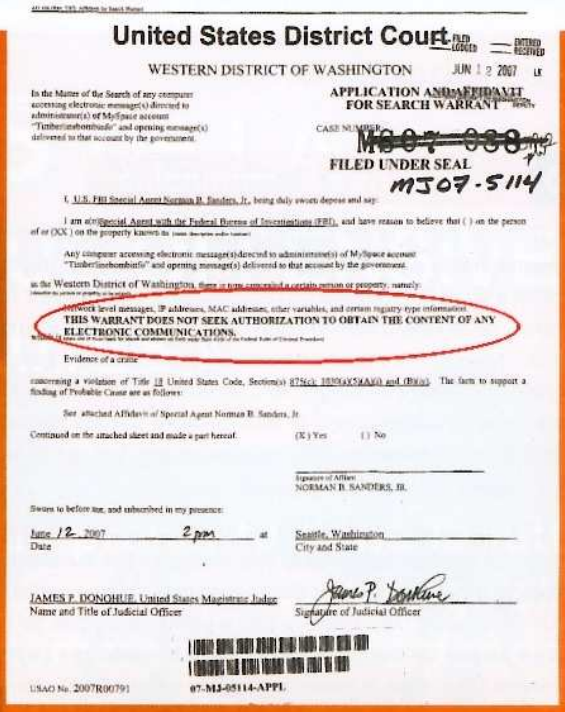
Sur un mandat délivré par le DoJ, l'envoi d'un CIPAV sur un ordinateur cible peut être décidé. Ce CIPAV fera envoyer par tout ordinateur l'activant des messages réseau contenant « l'adresse IP et/ou l'adresse MAC de l'ordinateur l'ayant activé, d'autres variables d'environnement et certaines informations d'enregistrement vers un ordinateur contrôlé par le FBI ». Ces informations consistent, entre autres, en :

- adresse IP réelle de la machine ;
- adresse MAC ;
- ports de communication ouverts ;
- liste des programmes en cours d'exécution ;
- système d'exploitation (type, version, n° de série) ;
- type et version de navigateur Internet ;
- encodage du langage ;
- nom de l'ordinateur, de la compagnie enregistrée, de l'utilisateur connecté ;
- URL visitées.

Ces informations ont pour but de permettre la localisation de l'ordinateur et l'identification de la personne qui l'a utilisé. La liste des données transmises indiquées en exemple dans le document du FBI ne semble être qu'un exemple du type d'informations récupérées lors d'une enquête. Rien n'indique que ce soit là les seules qui le soient.

Le CIPAV utilise des commandes Internet standards pour envoyer des messages, et/ou d'autres variables et/ou des informations sur la base de registres. La nature exacte de ces commandes, les processus utilisés, les capacités et la configuration des CIPAV sont classifiés. On n'en connaît actuellement que les fonctionnalités qui ont été utilisées dans le cadre de l'enquête « *Timberlinebombinfo* ». Le CIPAV, une fois installé, commence à surveiller l'ordinateur en enregistrant toutes les adresses IP auxquelles il se connecte. Il continue à transmettre des informations jusqu'à sa désactivation.

Les possibilités exactes de cet (ou de cette catégorie d') outil sont inconnues. Seules sont connues les caractéristiques révélées par la presse. Des capacités supérieures de ce produit n'ont pas été confirmées (accès au disque dur, fonctions d'enregistreur de frappe… comme pourrait le faire un cheval de Troie classique. Rien n'est sûr quant à l'existence d'autres outils de ce type. L'important, et c'est ce qui fait toute la différence avec « Magic Lantern », est que l'utilisation de ces « espiogiciels » (spywares) ou chevaux de Troie ou autres logiciels, se fait dans des conditions précises et strictement encadrées par le Département de la Justice, ce qui permet de limiter les risques de débordement. De nombreux rappels dans la déclaration du FBI précisent que les informations récupérées peuvent dans certaines conditions être transmises sur Internet et ne présentent pas forcément d'entrave à la vie privée (les variables d'enregistrement peuvent être transmises sur Internet lors d'un contrôle de validité d'une licence dans le cadre d'une mise à jour logicielle par exemple).



United States District Court

WESTERN DISTRICT OF WASHINGTON

JUN 12 2007

In the Matter of the Search of any computer accessing electronic message(s) directed to administrator(s) of MySpace account "Timberlinebombinfo" and opening message(s) delivered to that account by the government.

APPLICATION AND AFFIDAVIT FOR SEARCH WARRANT

CASE NUMBER: ~~M07-088~~

FILED UNDER SEAL

MJ07-5114

I, U.S. FBI Special Agent Norman B. Sanders, Jr., being duly sworn depose and say:

I am a(n) Special Agent with the Federal Bureau of Investigation (FBI), and have reason to believe that ( ) on the person of or (XX) on the property known as (name, description and/or location)

Any computer accessing electronic message(s)directed to administrator(s) of MySpace account "Timberlinebombinfo" and opening message(s) delivered to that account by the government.

in the Western District of Washington, there is now concealed a certain person or property, namely: (describe the person or property to be seized)

network level messages, IP addresses, MAC addresses, other variables, and certain registry type information

THIS WARRANT DOES NOT SEEK AUTHORIZATION TO OBTAIN THE CONTENT OF ANY ELECTRONIC COMMUNICATIONS.

which is (state one or more bases for search and seizure set forth under Rule 41(c) of the Federal Rules of Criminal Procedure)

Evidence of a crime

concerning a violation of Title 18 United States Code, Section(s) 875(c); 1030(a)(5)(A)(i) and (B)(ii). The facts to support a finding of Probable Cause are as follows:

See attached Affidavit of Special Agent Norman B. Sanders, Jr.

Continued on the attached sheet and made a part hereof. (X) Yes ( ) No

Signature of Affiant
NORMAN B. SANDERS, JR.

Sworn to before me, and subscribed in my presence:

June 12, 2007 Date — 2 pm — at — Seattle, Washington City and State

JAMES P. DONOHUE, United States Magistrate Judge
Name and Title of Judicial Officer

Signature of Judicial Officer

USAO No. 2007R00791 — 07-MJ-05114-APPL

2 *L'utilisation d'un CIPAV par le FBI a été abondamment commentée par la presse et les documents mis en ligne par le DoJ. Le mandat du DoJ précise expressément que si des données techniques peuvent être récupérées, l'interception du contenu des communications électroniques n'est pas autorisée.*

L'enquête « Timberlinebombinfo »

Les faits

Le 30 mai 2007, un message manuscrit indiquant la présence de bombes dans la *Timberline High School* conduit à l'évacuation du lycée. Du 4 au 8 juin, des e-mails contenant des alertes à la bombe conduisent à plusieurs évacuations de ce même établissement. En outre, l'e-mail du 4 juin annonçait la mise hors service du serveur de messagerie du lycée, qui subit effectivement une attaque par déni de service ce jour-là. Le 7 juin, 33 lycéens reçoivent des invitations, via leur compte MySpace d'un dénommé « timberlinebombinfo », leur demandant de placer un lien vers une page **http://bombermail.hyperphp.com**, en les menaçant, s'ils ne s'exécutent pas, d'associer leurs noms aux futures alertes à la bombe. D'autres encore ont reçu, via la messagerie instantanée de MySpace des invitations à discuter avec un dénommé « Timberlinebombinfo ». Ceux qui acceptaient étaient ensuite contactés via AIM (messagerie instantanée d'AOL). Ces communications instantanées semblent avoir tourné court dès que les personnes contactées demandaient des précisions sur les alertes à la bombe.

Les investigations préliminaires

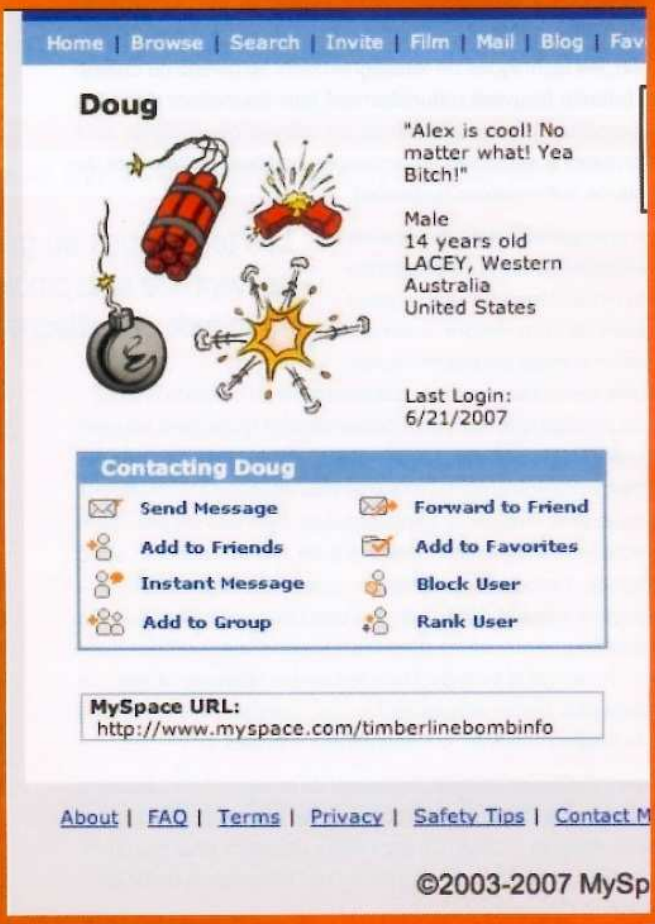
Dès le début, ces alertes ont déclenché des investigations menées par les administrateurs réseau du lycée (résolution de nom sur les origines des messages), la police locale et le FBI. Ces premières investigations ont montré que 5 adresses e-mail différentes et le compte MySpace « timberlinebombinfo » ont été utilisés pour diffuser ces alertes à la bombe. Des réquisitions auprès des fournisseurs d'accès Internet et des résolutions de nom sur les adresses IP utilisées ont permis de montrer que certains ordinateurs utilisés se trouvaient à l'étranger, en Italie. La coopération de la police italienne a alors permis d'identifier les ordinateurs compromis (Institut National de Physique Nucléaire, Sonic SRL).

L'utilisation du CIPAV

Au vu de ces premiers éléments, il est décidé de cibler le compte MySpace « timberlinebombinfo ». Un mandat est délivré le 12 juin, autorisant l'envoi d'un « CIPAV » sur tout ordinateur utilisé pour administrer ce compte d'utilisateur dans les 10 jours suivant sa délivrance. Ce logiciel est mis en place par messages électroniques adressés à l'administrateur du compte MySpace concerné, à partir d'un ordinateur contrôlé par le FBI, ordinateur qui sera destinataire des données recueillies.

À son activation, le CIPAV effectue une recherche sur le disque dur pour récupérer les variables d'environnement et d'enregistrement mentionnées plus haut. Cette recherche n'est effectuée qu'une fois par le logiciel. Celui-ci fonctionne ensuite comme un outil d'écoute (*pen register*) et enregistre les informations de routage et de destination des communications électroniques émises par l'ordinateur cible. Il enregistre les adresses IP, dates et heures des communications électroniques **sans leur contenu** et les retransmet vers le FBI pendant une durée de 60 jours.

Le 13 juin, l'activation du logiciel est réalisée et l'auteur des messages, un adolescent de 15 ans, identifié. Interpellé, il a reconnu être l'auteur des messages, avoir utilisé des ordinateurs compromis en Italie et avoir mené l'attaque en déni de service contre son lycée. Il a été condamné à 90 jours de détention dans un centre pour adolescents, 2 ans de probation assortis de restrictions sur l'utilisation d'Internet et des ordinateurs et a été exclu de son lycée.



3 *L'ex-étudiant avait créé un profil MySpace. C'est ce compte qui a été ciblé à l'aide du CIPAV. (Image : source : CNET Networks)*

Alors que ce sujet commence à être évoqué dans les armées de nombreux pays, il n'existe, du moins de manière ouverte, encore aucune réflexion sur la doctrine en particulier touchant à l'emploi de l'arme informatique dans un cadre offensif. Tous les éléments techniques disponibles permettraient pourtant de définir ce que pourrait être la conduite d'opérations de guerre informatique, en s'inspirant des techniques spécifiquement militaires. À titre d'exemple :

- L'emploi de l'artillerie peut être étendu aux attaques par déni de service (DoS ou DDoS).
- Le combat de l'infanterie et de la cavalerie, et leurs différentes techniques sont susceptibles d'inspirer des parallèles tout à fait pertinents sur la conduite d'opérations à l'aide de codes malveillants (l'articulation des groupes ou des sections d'infanterie peut être calquée par des codes k-aires [8] pour réaliser des missions de reconnaissance, d'appui, de couverture, d'assaut…). Plus spécifique à la cavalerie, la notion de « blindage de code » (total ou léger) autorise des comparaisons similaires.
- Les techniques du génie militaire, du train, et bien sûr des transmissions peuvent être sans problème transposées au monde des attaques informatiques.
- Enfin, les techniques de renseignements au niveau du champ de bataille trouvent naturellement leur équivalent dans les principales techniques mises en œuvre pour capter des informations sensibles (informations traitées) ou relevant du système (informations traitantes).

Le principal intérêt de ce parallèle réside dans le fait que la connaissance de l'art militaire classique peut permettre de comprendre comment des forces armées pourraient mettre en œuvre des ressources informatiques dans un contexte offensif – selon le principe que des outils différents sont finalement souvent utilisés selon une pensée unique – mais, d'autre part, ce parallèle peut permettre également d'imaginer des attaques informatiques beaucoup plus évoluées, sophistiquées, non pas au seul plan technique, mais dans leur conduite à un niveau tactique, voire stratégique. Force est de constater que – fort heureusement – les attaques « grand public » sont d'une pauvreté opérationnelle tactique frappante alors que les moyens disponibles sont potentiellement plus évolués. Dans le cas des attaques ciblées, ce n'est déjà plus systématiquement le cas, comme nous le verrons dans la section suivante, consacrée aux « brutes ».

Les techniques du génie militaire [...] peuvent être sans problème transposées au monde des attaques informatiques.

Avant d'effleurer ce sujet, il convient de replacer cette utilisation dans son contexte. La lutte informatique offensive (*Computer Network Attacks* – CNA) s'inscrit dans un cadre plus global de guerre de l'information (ou opérations de l'information (InfoOps – *Information Operations*)).

Les opérations de l'information

Les **InfoOps** sont des activités militaires coordonnées dans le domaine de l'information, visant à influer sur des informations et les systèmes d'information pour atteindre les effets désirés sur la volonté et les capacités d'adversaires et de tiers, en appui des objectifs de la mission, tout en soutenant les informations et systèmes d'information propres.

Elles touchent à tout ce qui, de près ou de loin, a un rapport avec la maîtrise de l'information. Le champ d'action est particulièrement vaste. Il repose sur 5 principaux domaines :

- les actions psychologiques : influer sur la perception et le comportement de l'adversaire ;
- les opérations de déception : amener les décideurs adverses à des prises de décisions fondées sur des informations erronées ;
- la sécurité des opérations : empêcher l'adversaire d'obtenir des informations sur les opérations amies ;
- la guerre électronique : maîtriser le spectre électromagnétique et en interdire l'utilisation par l'adversaire ;
- les opérations informatiques (CNO – *Computer Network Operations* – CNA – *Computer Network Attacks* – CNE – *Computer Network Exploitation*).

Il est essentiel de préciser qu'il s'agit là non seulement de la vision US, mais également de la sphère otanienne, élargie aux partenaires traditionnels de cette sphère. Certes, de ce point de vue, une certaine bipolarité s'est installée dernièrement : d'un côté, la vision US qui s'est imposée dans le monde occidental, d'un autre côté, la vision chinoise. Il est amusant de constater que la bipolarité OTAN/Pacte de Varsovie que l'on a connu dans le domaine de la guerre conventionnelle a cédé la place – tout en en calquant les principaux traits – à une bipolarité Ouest/Chine dans le domaine de la sphère digitale.

Plus récemment, d'autres domaines influant sur l'information sont venus s'y ajouter :

- relations publiques/relations avec les médias ;
- actions de coopération civilo-militaire ;
- contre-renseignement, contre-déception… où il s'agit de contrer les manœuvres adverses dans ces domaines.
- …

Il s'agit là de domaines complémentaires aux précédents qui viennent, dans la conception otanienne de l'InfoOps, en renfort des cinq points précédents, avec un niveau d'importance pratiquement égal.

Force est de le constater que, et toutes les crises récentes l'ont montré, la maîtrise de l'information est une part essentielle des opérations militaires modernes et l'information un enjeu-clé. Sous quelque forme que ce soit, tous les aspects de cet environnement sont sinon utilisés, du moins envisagés, comme c'est souvent le cas pour l'arme informatique.

Dans un contexte d'opérations d'information, la notion de « guerre informatique » comprend 3 volets :

- attaque (CNA – Computer Network Attack) : attaque des réseaux informatiques adverses (déni de service, injection de code malveillant...)
- exploitation (CNE – Computer Network Exploitation) : obtenir du renseignement.
- défense informatique (CND – Computer Network Defense) : la partie que tout le monde connaît et pratique : protéger ses réseaux contre les opérations adverses (CNA et CNE).

La problématique légale : un frein à l'emploi ?

La principale difficulté dans ce domaine vient des contraintes légales auxquelles doivent faire face les protagonistes. Les lois internationales qui s'appliquent (loi des conflits armés, charte des Nations Unies, traité de l'OTAN...), rédigées en des temps où la question ne se posait pas, n'abordent pas la question informatique. Il faut alors essayer de les interpréter et de les adapter à ce nouveau contexte. L'exercice se révèle difficile. Il faut en effet adapter des notions qui, déjà relativement complexes dans un cadre conventionnel, deviennent vite inextricables dès qu'il s'agit d'informatique.

Pour l'essentiel, ces questions rejoignent celles qui ont été posées par l'incident estonien (voir *Misc* 33 et 35) :

- Quand peut-on considérer qu'une attaque informatique est une agression armée ?
- Comment sont définis les objectifs militaires du point de vue de l'informatique ?
- Qu'en est-il des dommages collatéraux ?
- Comment (ou pourra-t-on) user de la légitime défense dans ce domaine ?

Les réponses à toutes ces questions restent encore à apporter, mais l'exemple du conflit actuel en Irak est révélateur de ces contraintes : si l'on en croit la presse [9], les États-Unis ont envisagé en 2002-2003 d'utiliser l'arme informatique contre l'infrastructure financière irakienne. Ces opérations ont été annulées compte tenu des « dommages collatéraux » estimés : une attaque informatique sur ces infrastructures aurait probablement également désorganisé les infrastructures financières... en Europe.

encadré 2

6 septembre 2007, raid de l'aviation israélienne en Syrie : un exemple de cyber-guerre ?

Le 6 septembre 2007, l'aviation israélienne entreprenait un raid en Syrie. Le succès de ce raid – même si les Israéliens ont, plusieurs fois dans l'histoire, montré leur maîtrise en la matière – a soulevé de nombreuses interrogations sur ses conditions d'exécution, l'aveuglement de la défense aérienne syrienne laissant beaucoup de monde perplexe.

Les conditions et le déroulement exacts de ce raid ne sont pas connus (le seront-ils un jour ?). Toutefois, certaines informations ont filtré quant à l'emploi par l'aviation israélienne d'un « système d'attaque réseau aéroporté » baptisé « Suter » **[10]**. Selon la revue *AviationWeeks* **[11]**, « cette technologie permettrait 'd'envahir' les réseaux de communications adverses, de voir ce que les senseurs adverses voient, et même de prendre le contrôle du système en tant qu'administrateur et de manipuler les senseurs de manière à ce qu'ils ne détectent pas les appareils en approche. »

...la seule limite c'est l'espace.

« Le processus passe par une localisation très précise des émetteurs ennemis, et par l'envoi vers ceux-ci de flux de données qui peuvent comprendre de fausses cibles et des commandes permettant de réaliser différentes 'activités', entre autres, d'en prendre le contrôle. »

Selon des analystes américains, cette attaque a, semble-t-il, comporté une part de brouillage classique. Ces analystes s'accordent à dire que le réseau électrique syrien n'a pas été attaqué, mais que la pénétration des réseaux a été réalisée à la fois par des moyens d'attaque électronique air-sol et par infiltration dans des réseaux informatiques point à point. D'après un spécialiste du renseignement, « il y a également eu des pénétrations à haut niveau, non tactique, qu'elles aient été directes ou des diversions et des usurpations, des moyens de commandement et de contrôle syriens, réalisées par le biais d'attaque réseau ».

« Le domaine des attaques réseau offensives et défensives est un des nouveaux domaines le plus intéressant », déclare Pinchas Buchris, directeur général du ministère israélien de la Défense, « je peux seulement dire que nous suivons ces technologies [d'attaque réseau] avec beaucoup d'attention. J'en doutais [de cette technologie]. Mais nous l'avons fait. Maintenant, tout a changé. »

« On a besoin de ce type de moyens », continue-t-il « Vous n'êtes pas responsable si vous ne la prenez pas en compte. Et si on peut réaliser ce genre de moyens, la seule limite [aux opérations de renseignement sophistiquées et aux opérations spéciales], c'est l'espace. »

Ce genre d'action pose toute la question de la numérisation de l'espace de bataille, de l'interdépendance grandissante avec les senseurs numériques qui en résulte, et de la vulnérabilité de ces senseurs aux cyber-attaques. Les Chinois ont une vision similaire, et considèrent que cette numérisation apporte certes des avantages, mais constitue également une grande vulnérabilité.

...les « brutes »...

Dès lors que les opérations militaires deviennent purement nationales, certains obstacles peuvent se lever et l'emploi de l'informatique en tant qu'arme n'est plus alors qu'une question de volonté nationale. Certains pays semblent d'ailleurs s'engager résolument dans cette voie. Les États-Unis, par exemple, semblent se doter de tels éléments juridiques : la directive présidentielle n°16 semble répondre à ces contraintes : elle lèverait les limitations imposées par les lois américaines pour l'utilisation de telles actions (si l'on en croit ce que la presse a révélé, cette directive étant secrète). Les USA sont par ailleurs en tête dans ce domaine en se dotant d'organismes comme le « *cyber-command* » : cet organisme a pour but d'assurer la maîtrise du cyber-espace et du spectre électromagnétique. Il reste encore à voir quels seront les moyens réels mis en place et quel en sera l'emploi. Ils semblent d'ores et déjà conséquents (les effectifs annoncés du cyber command sont de 5 000 à 10 000 hommes).

La démarche est illégale, un point c'est tout.

L'emploi de l'informatique dans les crises récentes a également permis de voir que si certains protagonistes s'astreignent à respecter certaines règles, le conflit déborde très vite sur le web et si, au niveau national, il y a souvent de fortes contraintes, il se trouve toujours des gens « bien intentionnés » qui, par motivation idéologique... pour afficher leur soutien à une cause... se livrent à une guerre à peu de frais sur l'Internet : les défacements de sites Internet en sont la partie la plus visible. Mais, ces attaques informatiques peuvent s'accompagner très facilement d'attaques plus sérieuses : déni de service [12], utilisation de virus informatiques [12]... pour, au final, arriver à s'échanger des menaces de mort [12]. L'origine réelle de telles attaques, voire leur commanditaire réel, restera, dans tous les cas, très difficile à établir (voir l'Estonie – *MISC* 33 et 35).

L'informatique, et les « attaques chinoises » qui ont défrayé la chronique il y a quelques mois l'ont mis en évidence, est redoutablement efficace dès que l'on touche au domaine de l'espionnage et au monde du renseignement. Il est bien évident que les informations ouvertes dans ce domaine sont très difficiles à obtenir. L'histoire est suffisamment riche en exemples montrant que les technologies les plus modernes ont été utilisées dans ce domaine, pour raisonnablement penser que, dans ce monde où il n'y a pas vraiment de règles, l'emploi d'un outil aussi puissant est maintenant incontournable [13].

De ce point de vue, dès lors que la barrière de la légalité a été franchie, rien ne distingue plus un agent clandestin (espion, mercenaire numérique...) d'un terroriste, si ce n'est, éventuellement, dans la phase finale de l'attaque, le terroriste cherchant souvent à rendre son action visible (destruction et propagande). La démarche est illégale, un point c'est tout. Mais, dans la pratique, on observe qu'un terroriste sera moins précautionneux qu'un agent gouvernemental clandestin, en situation d'illégalité. Le premier se moque des conséquences provoquées par la découverte de ses actions (il les revendique même), alors que le second engage sa responsabilité et celle, dans une moindre mesure, de son gouvernement, au moins de manière officieuse.

La principale conséquence est que, d'une manière générale, une « brute » ne passera pas par les réseaux pour monter son attaque ou du moins pas directement. Il est en effet essentiel qu'aucun élément technique ne permette de remonter jusqu'à lui, ou même de manière supposée, jusqu'à son pays. La plupart du temps, cela nécessitera soit de travailler directement sur le sol national de la victime ou bien sur le sol d'un pays tiers, qui, le cas échéant, endossera la responsabilité de l'attaque en cas de problème.

encadré 3

Les attaques contre l'infrastructure électrique, une réalité ?

Venant conforter les interrogations soulevées par les récentes expériences américaines montrant que l'on pouvait amener un générateur à s'autodétruire (voir *MISC* 35), des informations rendues publiques par Robert Donahue, analyste de la CIA, le 18 janvier 2008 à la Nouvelle Orléans lors d'une conférence organisée par le *SANS Institute* (*SANS Process Control and SCADA Summit 2008*), ont révélé que des attaques visant à désorganiser les réseaux d'alimentation électrique étaient peut-être plus proches qu'on ne pouvait le penser **[16]**.

« Nous avons des informations, de diverses régions à l'extérieur des États-Unis, rapportant des cyber-intrusions dans des infrastructures, suivies de demandes d'extorsion de fonds. Nous soupçonnons, mais sans pouvoir le confirmer, que certains des attaquants ont bénéficié de complicités intérieures. »

« Nous avons des informations selon lesquelles des cyber-attaques ont été utilisées pour mettre hors service de l'équipement électrique dans des régions extérieures aux États-Unis. Dans au moins un des cas, les perturbations ont entraîné une coupure

Pour illustrer tout cela, considérons l'attaque générique suivante, synthèse **prospective** inspirée par plusieurs cas réels démarqués – rencontrés lors d'expertises judiciaires – d'attaques de « brutes ». Bien sûr, toute ressemblance avec des faits réels ne serait que pure coïncidence.

Imaginons, un service de renseignement d'un pays Carmin souhaitant monter une mission d'espionnage illégal contre un système informatique S d'un pays « ami » Azur. Passer par les réseaux lui est impossible, car la loi de son pays interdit ce genre de pratique, et, « officiellement », ses chefs ne sont pas au courant. Se faire prendre est impensable.

Un agent Carmin est envoyé sur le sol national du pays Azur et/ou sur celui du pays Vert, allié d'Azur. Il ouvre plusieurs adresses email gratuites (de type Gmail). Pour cela, il passe par des connexions WiFi et un ordinateur portable volé, dans des lieux publics (hôtels, gare, voie publique...). Toutes ses dépenses sont payées en liquide, il change d'hôtel chaque soir, et ce, dans une ville différente du pays Azur ou du pays Vert. L'agent ensuite, par des techniques opérationnelles classiques, parvient à introduire un cheval de Troie sur les ordinateurs cibles visés du système S **[14]**. Ce cheval de Troie, indétectable, collecte juste quelques données critiques (*login* et mot de passe des utilisateurs) et les renvoie, sous une forme anodine (au moyen de stéganographie ou par des techniques de cryptographie malicieuse **[15]**, vers l'une des adresses email, choisie au hasard, que l'agent Carmin a créées. Ce dernier n'a plus qu'à relever régulièrement ses boîtes et collecter les données qui lui permettront ensuite, selon le même protocole discret, de se connecter sur le système S et de voler les données sensibles recherchées.

En cas d'enquête – la fuite de documents est constatée – l'enquête ne révèle que très peu d'éléments techniques, lesquels ne permettent pas de remonter à l'auteur de l'attaque et encore moins à son pays d'origine.

Il est alors intéressant de réfléchir que si le pays Carmin souhaite mener son attaque et incriminer un pays tiers – à tort, mais cela peut être précisément le but réel d'une attaque –, il s'arrangera au contraire pour laisser des traces permettant de remonter à ce pays tiers, et laisser à penser que l'attaque a été menée par ce pays. Les possibilités techniques et les scénarii possibles permettent des variations infinies sur ce thème. Cela explique en outre toute la difficulté, pour ne pas dire l'impossibilité d'identifier exactement l'auteur réel d'une attaque, dans un tel contexte.

...et les « truands »

Les truands, quant à eux, sont les acteurs dont la motivation est l'appât du gain. Ce sont les mieux – ou les moins mal – connus. Ils ont été largement présentés dans les articles précédents (voir *MISC* 35). Le plus souvent, le niveau des attaques n'est pas très évolué. Point n'est besoin. C'est la loi du nombre qui commande. Une attaque en masse – par exemple de type *phishing* – ou ciblée – attaquer un serveur de commerce en ligne pour en récupérer les milliers d'identifiants bancaires qu'il renferme – sera rentable par nature : même une faible proportion de victimes finales permettra d'engranger des bénéfices substantiels.

Tous ces réseaux sont hyper cloisonnés chaque groupe a un rôle bien défini.

À titre d'illustration, décrivons le mode opératoire classique d'une organisation mafieuse chinoise réalisant des opérations de type *carding*. Des milliers d'identifiants complets (incluant les codes PIN) sont volés. Des clones de cartes sont ensuite réalisés. Des pauvres hères chinois sont envoyés en Europe – leur famille sert de « caution » – pour réaliser des achats de produits de luxe et les ramener en Chine où ils seront ensuite revendus. Lorsque la justice fait son œuvre, une fraction non négligeable parvient à passer entre les mailles de la police, ce qui rend ce trafic néanmoins juteux.

Tous ces réseaux sont hyper cloisonnés – chaque groupe a un rôle bien défini. Mais l'autre caractéristique tient au fait que, le plus souvent, la jungle des législations nationales leur permet de relativement bien échapper aux poursuites. Basés dans des pays d'où l'on m'extrade que rarement, où la police est bienveillante (voir *MISC* 35), les coopérations internationales en matière de police et de justice sont encore embryonnaires, voire illusoires concernant certains pays.

Pour conclure avec les truands, disons simplement que l'avenir risque d'être plus sombre avec l'élévation du niveau technique des attaques. Et comme en témoigne un exemple récent (voir encadré « Les attaques contre l'infrastructure électrique »), les « truands » pourraient bien jouer dans la même cour que les « brutes » et considérer des cibles de même nature.

affectant plusieurs villes », a poursuivi M. Donahue. « Nous ne savons pas qui a exécuté ces attaques, ni pourquoi, mais elles ont toutes été réalisées par des intrusions venant de l'Internet. »

Les villes et pays concernés, la date et la durée de ces perturbations n'ont pas été précisés.

La CIA, par l'intermédiaire d'un de ses porte-parole, George Little, a fait savoir que « toute l'information qui peut être rendue publique l'a été. Elle avait pour seul but de souligner les défis que posent des cyber-intrusions potentielles. »

encadré 4

LIO – un exemple de scénario possible

Une attaque informatique d'importance et efficace nécessite quand même quelques préparations. Des dénis de service peuvent rapidement être lancés à l'aide de *botnets* sans qu'une longue préparation soit nécessaire, mais les effets durent rarement plus longtemps que l'attaque elle-même. L'Estonie l'a bien montré.

Imaginons le scénario suivant :

Un pays X manufacture (ou des entreprises d'un pays Y ont délocalisé dans ou sous-traitent à des entreprises du pays X...) des produits x. Actuellement, de l'informatique embarquée se retrouve dans de nombreux matériels de consommation (presque) courante. Il est alors très simple pour le pays X d'introduire dans le logiciel embarqué, ou dans le matériel x, des fonctionnalités cachées, mises en sommeil et n'attendant que d'être activées. Implantée dans le matériel à l'origine, spécialement conçues dans un tel but, ces fonctionnalités n'ont que peu de chances (pour ne pas dire aucune) d'être détectées.

Il ne reste plus qu'à attendre que le produit x soit commercialisé à grande échelle (chose relativement simple si l'on choisit bien sa cible) pour disposer, un peu partout dans le monde, de véritables bombes informatiques n'attendant que d'être activées.

Que survienne un jour une crise grave impliquant le pays X, pour que celui-ci décide d'activer ces fonctionnalités cachées : il pourrait ainsi semer, à peu de frais, un véritable chaos informatique. Pour sortir du domaine étatique, une telle fonctionnalité injectée dans les produits par une organisation plus mafieuse pourrait lui permettre de mener insidieusement une véritable guerre informatique dont les bénéfices peuvent être conséquents.

Il est vrai que si commercialiser des produits divers avec du code malveillant présent dès l'usine n'est pas une chose courante, l'actualité a montré que cela n'était pas aussi rare que cela :

Conclusion

Si les « bons » peuvent être clairement identifiés, leur volonté de respecter les législations en vigueur leur imposant certaines contraintes, les autres tendent à se mélanger, les derniers vendant leurs services aux seconds. Des rapports américains **[17]** le soulignent depuis plusieurs années : la distinction entre les organisations terroristes et mafieuses sur Internet tend à se faire de plus en plus difficile. Les premiers n'hésitent pas à recourir aux services ou aux méthodes des seconds, généralement pour financer d'autres actions. Quant aux « brutes », ils représentent la « face cachée » de la LIO. Et, à moins d'erreurs grossières de leur part, il est peu probable que leurs activités ne soient jamais révélées. Au fond, un scandale de type « *Rainbow Warrior* numérique » est tout sauf probable. La technique opérationnelle a tellement de moyens de faire porter le chapeau soit aux « bons », soit aux « truands ». C'est Sergio Leone au pays du virtuel.

Notes & Références

[1] **http://www.securityfocus.com/news/354**

http://ca.com/fr/securityadvisor/pest/pest.aspx?id=50498,

http://www.trust-us.ch/cryptome/01-Cryptome-061213/dirty-secrets2.htm

http://www.trust-us.ch/cryptome/01-Cryptome-061213/dirt-safrica.htm

Le manuel complet du logiciel DIRT est disponible sur : **http://cryptome.quintessenz.org/mirror/DIRTManual2_2clr.pdf**

[2] On pourra consulter, entre autres liens :

http://epic.org/privacy/carnivore/foia_documents.html

http://www.rumormillnews.com/cgi-bin/archive.cgi?noframes;read=15391

http://usgovinfo.about.com/library/weekly/aa121401a.htm

[3] La déclaration sous serment de l'agent spécial du FBI ayant procédé à l'utilisation de ce logiciel est accessible sur Internet : **http://blog.wired.com/27bstroke6/2007/07/fbi-spyware-how.html**

[4] **http://www.techworld.com/security/news/index.cfm?newsID=10446**

http://www.lefigaro.fr/international/20070831.FIG000000267_l_espionnage_informatique_fait_debat_outre_rhin.html

http://www.iht.com/articles/ap/2007/08/31/europe/EU-GEN-Germany-Trojan-Horses.php

http://www.lefigaro.fr/actualites/2007/12/15/01001-20071215ARTFIG00124-bientot-des-mouchards-de-police-sur-les-ordinateurs-.php

[5] En 2001, le *Washington Post* avait indiqué que Symantec (Norton Antivirus) et Network Associates (McAfee) auraient répondu favorablement à une requête du FBI leur demandant de faire en sorte que Magic Lantern ne soit pas détecté par leurs logiciels. Les deux sociétés ont par la suite démenti.

IBM, Yamaha, Activision (1999), Creative (2005), Apple (2006), Seagate/Maxtor (2007), les cadres photos numériques (2007-2008) sont là pour le prouver (sans parler de tous les magazines dont les CD-ROM sont certifiés testés contre tous les virus connus) !

Toute ressemblance avec des pays existants ou ayant existés serait purement fortuite. Ce scénario ne sort que de notre imagination !

Pour des exemples avérés :

- **http://www.cnn.com/TECH/computing/9904/08/aptivirus.idg/index.html**
- **http://www.timesdaily.com/apps/pbcs.dll/section?category=NEWS&template=wiki&text=CIH_virus**
- **http://www.lesnouvelles.net/articles/virus/lecteurs-mp3-creative-infectes**
- **http://www.apple.com/fr/support/windowsvirus/**
- **http://www.sophos.com/pressoffice/news/articles/2006/10/ipod-ships-with-virus.html**
- **http://www.pcinpact.com/actu/news/39993-Seagate-Maxtor-Personal-Storage-3200-virus-t.htm**
- **http://www.upi.com/NewsTrack/Science/2008/02/18/digital_frame_virus_traced_to_china/9409/**
- **http://www.sfgate.com/cgi-bin/article.cgi?f=/c/a/2008/02/15/BU47V0VOH.DTL**
- **http://www.pcinpact.com/actu/news/41468-cadre-photo-virus-infection-insignia-best-bu.htm**
- **http://www.f-secure.com/v-descs/cih.shtml**
- **http://www.heise.de/newsticker/meldung/23498**

Pour d'autres possibilités :

- **http://www.silicon.fr/fr/silicon/news/2007/09/03/rootkit-de-sony-une-incroyable**
- **http://www.pcinpact.com/actu/news/Les_premieres_voitures_infectees_par_un_virus_.htm**
- **http://www.rfidvirus.org/index.html**

[6] **http://news.zdnet.com/2100-1009_22-6197020.html**

http://news.zdnet.com/2100-1009-6196990.html

[7] Il convient de faire attention lors de la consultation de traductions françaises de documents étrangers traduits en anglais, puis en français. On peut souvent se rendre compte que la notion de « guerre de l'information » est improprement traduite par « guerre informatique ».

[8] Voir FILIOL (E.), *Techniques virales avancées*, Collection IRIS, Springer Verlag, chapitre 3, 2007.

[9] **http://archive.newsmax.com/archives/articles/2003/3/12/134712.shtml**

[10] Selon AviationWeek, (**http://www.aviationweek.com/aw/generic/story_generic.jsp?channel=awst&id=news/02145p04.xml**), le système Suter, dont la dernière version aurait été testée en Irak et en Afghanistan, aurait compris trois évolutions :

- Suter1 permettrait de « surveiller » ce que voient les senseurs ennemis.
- Suter2 permettrait en plus de prendre le contrôle du réseau adverse en tant qu'administrateur et de manipuler les senseurs.
- Suter 3 permettrait de s'attaquer aux liens de communication de cibles plus critiques (lanceurs de missiles balistiques ou sol-air).

Les auteurs de cet article concluent en indiquant que les programmes militaires relatifs aux opérations d'informations et aux attaques informatiques sont maintenant considérés comme parmi les plus sensibles, surpassant même dans ce domaine les dernières techniques de furtivité pour les aéronefs.

[11] **http://www.aviationweek.com/aw/generic/story_channel.jsp?channel=defense&id=news/aw100807p2.xml**

http://www.aviationweek.com/aw/generic/story_channel.jsp?channel=defense&id=news/aw112607p2.xml

http://informationdissemination.blogspot.com/2007/10/electronic-war-in-iaf-strike-in-syria.html

[12] Pour ne donner que quelques exemples (on pourrait donner des pages de liens) :
http://www.landfield.com/isn/mail-archive/2003/Jan/0008.html

http://www.eurasianet.org/departments/insight/articles/eav020807a.shtml

http://www.cnn.com/TECH/computing/9905/12/cyberwar.idg/

[13] Outre les attaques chinoises dont on trouvera les références dans le numéro 33 de MISC, on peut également consulter l'ouvrage de Nima Zamar : *Je devais aussi tuer.*

[14] En règle générale, l'agent Carmin devra éventuellement et temporairement agir sur le sol du pays Azur, selon la cible S envisagée. Certains lecteurs argueront que cette étape est probablement la plus difficile à mener – notre chère adjudant-chef Lansiaux Isabelle nous dirait avec un D comme « Dans tes rêves » – et pourront mettre en question sa validité. L'expérience montre que, au contraire, c'est certainement la phase la plus facile. Le meilleur exemple est sans aucun doute le cas d'espionnage qui a frappé la chancellerie allemande en août 2007, à côté de très nombreux autres cas. Pénétrer un système n'a jamais été aussi facile.

[15] Voir FILIOL (E.) et RAYNAL (F.), « *Malicious Cryptography* », CanSecWest'08, Vancouver, mars 2008.

[16] **http://www.sans.org/newsletters/newsbites/newsbites.php?vol=10&issue=5**

[17] *Report for Congress (RL33123) – Terrorist capabilities for cyber-attacks: overview andpolicy issues, Jan. 22 2007.*

Michel Iwochewitsch – mi@stratinternational.com

PETIT MÉMO D'IDENTIFICATION DES FAILLES À L'USAGE DU PRÉDATEUR INFORMATIONNEL…

Le choix du titre est volontairement accrocheur ! Pourquoi ? Parce que cette phase d'identification est ESSENTIELLE dans le métier du prédateur, et donc lorsque nous testons la sécurité informationnelle. Nous nous proposons dans le cadre de ce court article de présenter une approche opérationnelle exploitable pour reproduire le comportement des prédateurs lors des « tests de sécurité informationnelle » réalisés en accord avec les entreprises clientes.

mots clés : ***identification de failles / approches offensives / renseignement / espionnage industriel***

Avant d'aborder le sujet proprement dit, quelques limites et rappels sont cependant nécessaires. Premièrement, l'approche présentée fait appel à un ensemble de techniques – légales ou illégales – qui ont une vocation pédagogique !

L'auteur rappelle qu'il condamne fortement le recours à ces approches en dehors du cadre fixé : les tests de pénétration à vocation d'audit de sécurité informationnelle ! Néanmoins, nous considérons que « seul un braconnier peut devenir un bon garde-chasse » : il est donc nécessaire de comprendre les mécanismes déployés par l'opposition.

L'auteur rappelle qu'il condamne fortement le recours à ces approches en dehors du cadre fixé : les tests de pénétration à vocation d'audit de sécurité informationnelle !

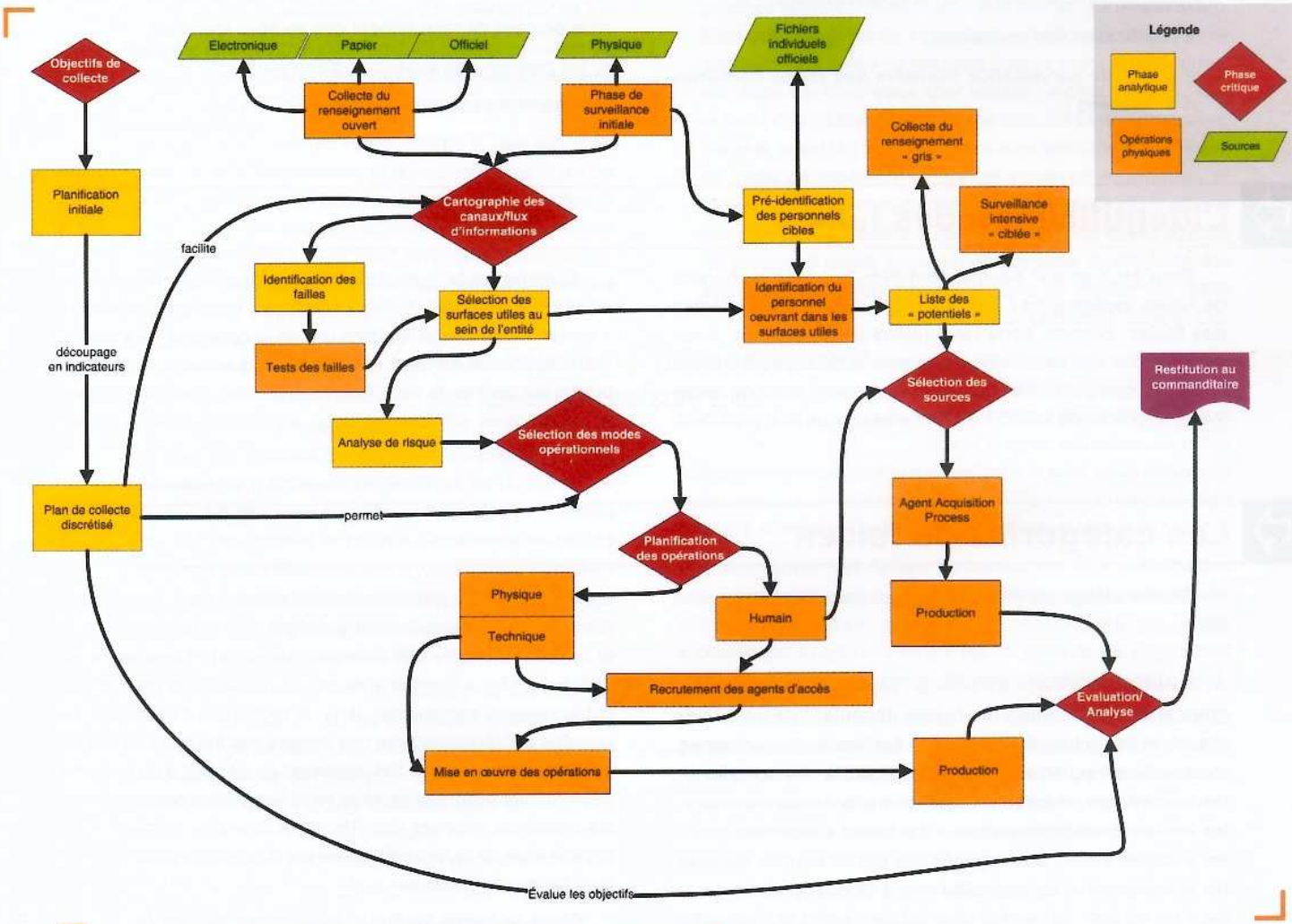
Pour ceux qui seraient néanmoins tentés par « l'aventure illégale », l'auteur tient à rappeler qu'un ensemble de textes punit ces aventures ! De la Directive Européenne 95/46/CE sur la vie privée, aux nombreuses législations des pays européens (la collecte d'informations confidentielles – y compris sous forme orale – est condamnable), en passant par l'EEA américain de 1996, **le cadre juridique de l'emploi abusif de ces méthodes est limpide : vous finirez en prison sans passer par la case départ !**

Un point sur les définitions : par prédateur informationnel, nous entendons tout acteur collectant de l'information dans un but offensif. Notre définition personnelle intègre donc le *social engineer* ; les hackers illégaux soutenus ou non par des groupes criminels transnationaux [1] ; le traitant [2] des services étatiques et/ou d'organismes privés spécialisés dans le renseignement industriel illégal ; les opérationnels en guerre de l'information, désinformation, *perception management* et opérations psychologiques ; et les arnaqueurs en tous genres [3].

Par failles, nous entendons l'ensemble des failles pouvant être exploitées par un prédateur ! Notre typologie (cf. Figure n°2, page suivante) nous est propre, et diffère d'autres opérationnels en sécurité. Ainsi, nous n'avons pas adopté le principe général du support de l'information (électronique, papier, etc.) comme critères de catégorisation. **En effet, notre constat rejoint celui des opérationnels du renseignement offensif : la forme de l'information n'a pas d'intérêt majeur pour ce dernier !** Pour illustrer notre constat, prenons l'exemple d'un mot de passe. Si l'opérationnel désire l'obtenir, que ce dernier soit obtenu par pénétration informatique, *social engineering*, élicitation lors d'une soirée ou sur un post-it déchiré dans une poubelle... Le résultat sera le même : l'information obtenue est UTILISABLE !

Pour illustrer notre propos, nous adopterons un « fil rouge » au long de cet article : la société « *NoLuckInLife* », – NLIL pour les intimes – œuvrant dans la confection textile (pour limiter les « vocations tardives») – qui comprend une direction centrale (HQ), un service commercial (SC) plus itinérant, et un site industriel (SI) et, a recours à la sous-traitance. Imaginons également que NLIL soit d'origine américaine (l'EEA étant un « plus » obligeant à renforcer les OPSEC [4]), plus précisément localisé en Californie (pour son climat agréable évidemment, et pour la rigueur de ses textes juridiques, toujours dans l'objectif de limiter les « vocations »).

Toujours dans le cadre de notre exposé, imaginons que NLIL ait un concurrent US, « *NoMoreLuckInLife* (NMLIL) », qui concurrence directement NLIL sur ses lignes de produits. Alpha, le dirigeant de NMLIL rencontre par un beau jour ensoleillé (la Californie !) dans son *country-club*, Mr X spécialiste du renseignement industriel. Après un échange tout empreint de politesse, Alpha explique à Mr X le risque concurrentiel de NLIL pour ses activités. Mr X propose alors à Alpha de mettre en place une opération de collecte agressive et illégale pour « documenter » les actions que pourrait prendre Alpha contre NLIL [5]. Une fois les conditions financières acceptées, Mr X se met au travail avec son équipe...



Le process d'identification lors d'une opération offensive d'accès à l'information

Le process

Quid du process qu'utilisera l'équipe de Mr X ? Plusieurs modèles existent en fonction de l'origine et de la formation des opérateurs. Nous proposons ici un modèle simplifié qui nous sert de référence lors des tests de « sécurité informationnelle » (cf. Figure n°1, page précédente).

Les grandes étapes sont :

- la collecte de l'ensemble du renseignement ouvert pour « dérouler la pelote de laine » [6] ;
- la phase de surveillance initiale destinée à pré-identifier les cadres et/ou personnels importants (cf. en infra)[7] ;
- la cartographie des canaux/flux ET l'identification des failles potentielles ;
- la sélection des « surfaces utiles », et les éventuels tests des failles ;
- l'analyse de risque et d'intérêt de la surface utile et de la typologie d'information présente pour l'opération ;
- la planification des opérations ;
- la phase de surveillance intensive des cibles humaines sélectionnées ;



2 *Les catégories de failles retenues*

- le recrutement des agents d'accès pour les opérations techniques et physiques ;
- la mise en œuvre des opérations physiques et techniques ;
- le process de recrutement des sources identifiés ;
- les process d'évaluation et d'analyse ;
- la restitution au commanditaire

L'identification des failles

Pour Mr X et son équipe, tout débute par la recherche de failles exploitables ! Il existe de multiples composantes des failles. Comme nous l'indiquions en préambule, nous n'adopterons pas l'approche traditionnelle du support comme critère de catégorisation. La raison essentielle étant de rester « au plus proche du terrain » et d'imiter la logique d'un prédateur.

Comme toute typologie, notre approche présente de nombreux défauts ! Ainsi, il n'est pas toujours évident de « ranger » certaines failles dans une seule catégorie. Néanmoins, notre approche en quatre familles nous semble cohérente, puisqu'elle permet de déployer des approches méthodologiques et techniques différentes pour en faciliter l'exploitation.

Les catégories de failles

Quatre catégories ont été retenues dans notre approche (cf. Figure n°2) :

Failles opérationnelles : par failles opérationnelles, nous entendons toutes les failles découlant des activités régulières de l'entreprise. Il en existe évidemment des dizaines, mais les plus fréquentes sont : la prédictibilité des opérations ; les procédures, la mise en œuvre réelle des procédures ; les faibles procédures en cas « d'incident » informationnel ; les « traces » liées aux voyages, cartes de crédits, factures de téléphones ; l'*outsourcing* et/ou la multiplication des lieux de travail ; la gestion des contractants ; et l'universel « trop peu d'informations » qui ne permet pas de protéger correctement les informations importantes dans la structure (on ne peut consciemment protéger une information, que si on connaît la valeur de cette dernière) ; l'existence de directions plus « fragiles » de par leurs fonctions (marketing, public relation, commercial, RH) ; la recherche d'efficience qui entraîne par regroupement des acteurs des fragilités en cas de pénétration (un ex : les RH disposent en général d'un système informatique propre et protégé dans les grands groupes, mais les membres internes des RH ont accès à ce dernier) ; une organisation de la sécurité basée sur la préconception erronée de « l'attaquant extérieur », etc.

Dans le cadre de NLIL, la majorité de ces failles est présente ! Mr X étudiera particulièrement :

- Le fonctionnement du service comptable (« là où l'argent change de mains, l'information passe » : un des axiomes de son métier).
- L'importance de la sous-traitance dans le processus de fabrication : comme nous sommes sur le sol US, Mr X va commencer par appliquer une des règles essentielles de la collecte ouverte : « *do your homework* ! » (ouvert, semi-ouvert, administratif, etc.) au niveau fédéral, régional, étatique, local ! Ainsi, à titre d'exemple, pour la sous-traitance, Mr X fera réaliser une recherche sur la base PierImport lui permettant d'accéder aux copies des manifestes des bateaux ayant transportés les matières premières et/ou transformées de NLIL ! Il aura également accès aux noms des expéditeurs (et donc souvent aux sous-traitants) et des logisticiens. Avec le nom des logisticiens, Mr X utilisera une « *shell company* » [8] au Delaware pour demander des devis à ces sociétés sur des trajets identiques à NLIL : il pourra ainsi estimer les frais logistiques de NLIL.
- La surveillance initiale : elle comprendra de nombreuses actions pour lesquelles Mr X utilisera des agents de supports recrutés par ailleurs. Quelques exemples de la surveillance sur NLIL :
  - Recruter un détective privé en Californie pour suivre au GPS les camions des logisticiens lors des débarquements du port (légal en Californie, illégal dans d'autres zones) ;
  - Établir les flux extérieurs intéressant Mr X : par exemple, identifier les restaurants les plus courants pour la pause de midi, les bars favoris lors des soirées, le degré de sécurisation des accès (anonyme, visiteurs non annoncés, visiteurs avec RDV, employés temporaires, etc.), identifier les sociétés de livraisons alimentaires (une simple copie de la facture permet de savoir qui reste entre midi et 13h) [9], comprendre les horaires des employés et des femmes de ménage, les routines de sécurité, l'emplacement des poubelles, la fréquence de passage de la voirie, les camions (et les noms (sur les bâches) des logisticiens avec la fréquence et le volume, l'existence de stocks extérieurs. Ces données sont typiquement couplées avec celles disponibles en sources ouvertes (Google earth, photos satellite, photos aériennes, documents légaux (permis de construire, etc.), etc.).
  - Établir une surveillance du parking du HQ afin de détecter les cadres et les employés (habillement, horaires, type de véhicule). Passer ensuite les plaques dans les services de l'État pour identifier les noms des individus (légal aux USA). Identifier dans les BdD des délivrances de permis de conduire (support important compte tenu de l'absence de CNI) pour retrouver les adresses. Obtenir des copies des *credit reports*, des CB, etc. (de légal à parfaitement illégal). Le parking permet également de vérifier certaines caractéristiques humaines (ex : qui roule dans un véhicule au-dessus de ses moyens, qui a des enfants et quel âge ont-ils (siège auto) ? Degré de propreté du véhicule (intérieur/extérieur) ? Degré d'entretien et âge du véhicule (signe éventuel de difficultés financières) ? Etc.)
- L'organigramme interne : plusieurs solutions s'offrent à Mr X qui vont du vol pur et simple d'une copie (chez NLIL à l'accueil, chez l'imprimeur si fichier électronique, etc.), à la récupération d'un exemplaire auprès d'un ex-employé, en passant par le recours à des sous-traitants spécialisés des cabinets de chasse de tête, qui sur le sol US vous proposent de récupérer un organigramme à jour d'une société...
- Les flux des cartes de crédits, déplacements des cadres, etc. Pourquoi ? Car la combinaison des deux permet de savoir qui se rend où ? Et dépense quoi ? Ainsi, la classe de l'avion – comme le type d'hôtel – sont synonymes d'importance dans l'entreprise. Le relevé des CB pour les repas d'affaires permet de savoir où ces derniers ont eu lieu et le nombre estimatif de convives. Un simple repérage géographique des acteurs principaux du secteur d'activité est ensuite possible et donc l'identification des clients, partenaires, etc. Il est possible aussi de « pousser » plus loin et d'obtenir par elicitation les noms de certains cadres présents au repas, etc. Mais me direz-vous cher lecteur : « c'est illégal » ! Oui et non ! Oui en Europe, pas dans tous les États américains. De plus, le sol US regorge de sociétés spécialisées réalisant ce type de recherches contre espèces sonnantes et trébuchantes. L'accès au système SABRE de réservations des places est possible à partir de n'importe quelle agence de voyage. Il existe sur Internet une base de données des appareils privés qui permet ensuite de vérifier les plans de vol sur certaines zones. Etc.

À ces failles universelles se rajoutent des failles plus personnelles...

Failles humaines : Nous ne pourrons dans le cadre de cet article intégrer l'ensemble des failles humaines exploitables. Pour approfondir le sujet, nous renvoyons notre lecteur à un article paru dans MISC [10], et pour les biais cognitifs et leurs impacts, aux sites de références Internet [11].

Néanmoins, les failles humaines les plus universelles concernent : la personnalité [12] ; les biais cognitifs/heuristiques ainsi que les grandes lois d'influence [13] ; les failles « classiques » répertoriées par les services de renseignement (un exemple, MICE : *Money, Ideology, Compromission, Ego*).

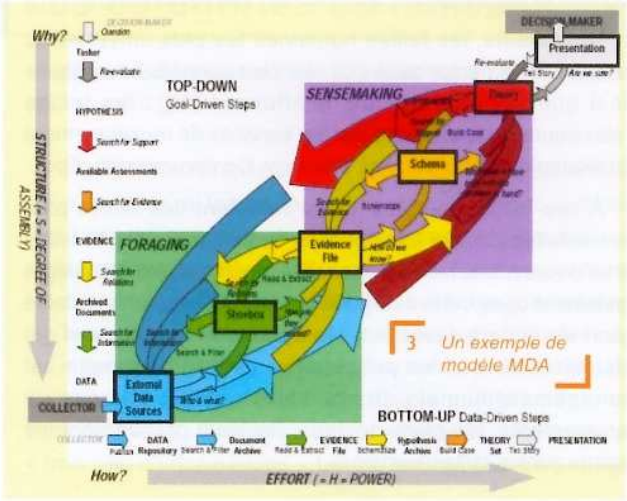
À ces failles universelles se rajoutent des failles plus personnelles comme les « phases de déstabilisation », telles qu'un divorce, une faillite personnelle, un décès, etc. On intègre également dans cette catégorie les failles environnementales ayant un impact direct sur « le moral des troupes » et qui cristallisent les failles précédentes. À titre d'exemple, un management humain de mauvaise qualité, des cas de harcèlements, les plans sociaux, facilitent grandement les actions sur les failles humaines !

Dans le cas de NLIL, certaines de ces informations personnelles sont facilement « collectables » en sources ouvertes ! Quelques exemples : registre légal pour les divorces, sources ouvertes pour les plans sociaux, fichiers divers et variés aux USA sur l'adhésion syndicale, religieuse, associative, etc. Le reste du travail consistera à profiler avec le soutien d'une surveillance intensive ciblée les individus sélectionnés comme « potentiels »... L'un des avantages directs de cette méthode est de pouvoir identifier des réseaux autres que professionnels pour entrer en contact avec les sources potentielles. L'autre, étant de pré-catégoriser les individus identifiés en fonction de failles potentielles.

Dans le cas de NLIL, certaines de ces informations personnelles sont facilement « collectables » en sources ouvertes !

D'autres failles naturelles sont souvent présentes dans les entreprises : l'absence de *chinese wall* entre les directions, les services administratifs ; le terrible « radio lavabo » ; l'« îlot RH » qui par ses prérogatives « bloque » des informations importantes (ex : un ex-employé d'un concurrent intégré dans un process à haute valeur ajoutée sans que le chef de projet ne le sache compte tenu de la sacro-sainte protection du CV de l'employé par les RH).

Failles physiques : très nombreuses, ces dernières regroupent les failles découlant d'une situation physique ! Parmi les plus fréquentes, on trouve : l'absence ou le faible contrôle des points d'entrée (exemple typique : le parking extérieur) ; une localisation physique facilitant les opérations techniques (ex : nous connaissons l'existence d'une salle de réunion réservée au conseil d'administration dans un grand groupe européen, dont les fenêtres donnent sur une annexe technique d'un service de renseignement étranger) ; le gardiennage de faible qualité (formation/motivation) ; le voisinage lorsque les *open space* sont à peine « séparés » par une demi-paroi ; les serrures fragiles et/ou sous-exploitées ; les bureaux mal rangés ; les stocks extérieurs de matériels ; le stockage électronique *outsourcé* (ex : archives confiées à une société extérieure) ; le *waste archeology* (fouille des poubelles) des déchets (rappelons que, même *shreddés*, les documents peuvent être reconstitués si le *shreddage* ne respecte pas certaines normes de qualité) ; les bureaux mal rangés ; l'absence de procédure (ou son non-respect) de sécurité des PC (*passwords*, écran de veille, etc.) ; les points fragiles naturels (ex : le copieur/imprimante et son disque dur) ; et la portabilité des équipements électroniques actuels (depuis l'apparition de ces outils, le volume de documents « volés » est en nette augmentation dans les cas identifiés – un ex : le cas célèbre Ericsson où l'opérateur a emporté « l'équivalent papier » d'une dizaine de 33 t sur des CD-ROM au fil de sa mission), etc.



3 *Un exemple de modèle MDA*

Il est impossible de présenter toutes les failles potentielles d'une société comme NLIL dans cet article. À titre d'exemple, il est facile d'entrer physiquement dans le HQ (immeuble hébergeant diverses sociétés) en « faisant acte de présence » sur la porte latérale réservée aux fumeurs ostracisés aux USA. Une présence fréquente accompagnée d'une politesse de bon aloi (ex : tenir la porte latérale ouverte) permet rapidement d'être dans le « groupe » et donc de rentrer avec les autres... L'intérêt essentiel est que ce type de porte latérale est peu protégé par les systèmes conventionnels de contrôle d'accès (en général pas du tout, le concept de « porte latérale » n'étant pas procédurisé !). Au pire, l'opérateur devra « fabriquer » un badge ressemblant. Sur le site industriel, une visite officielle avec RDV, permet de placer son véhicule sur le parking ! Notre pauvre opérateur ne comprenant aucunement les indications du poste de contrôle, se garera au plus loin de son point de RDV ! Il devra donc remonter à pied ! Prendre des raccourcis (ex : les portes ouvertes des ateliers en été pour des raisons de chaleur), etc.

Failles techniques : par « technique », nous entendons l'ensemble des failles pouvant être exploitées par un mode technique. Ces derniers allant de la pénétration informatique, à la sonorisation [14] d'un lieu, en passant par la surveillance électronique, le Tempest, les EMP, etc. Les principales familles de failles dans cette catégorie sont : les failles logicielles, erreurs de configuration, des outils informatiques ; la gestion des passwords ; le stockage des archives ; les canaux de transmission (interne/externe) de l'information ; les réseaux et points d'accès déployés (ex : l'accès itinérant des commerciaux qui est protégé par une *smartcard* et une sécurisation des réseaux, mais qui n'intègre que peu le risque qu'une smartcard officielle soit détournée de son usage légitime) ; les EMP [15] et approches Tempest [16] dont les coûts de fabrication ont fortement diminué ces dernières années ; les techniques de sonorisation des lignes (fax/voix), des lieux qui vont de l'ultra-sophistiqué réservé aux services officiels, à des outils simples à prendre en main et suffisant pour de nombreux cas de renseignement industriel.

De très nombreuses approches techniques existent au-delà même des actions informatiques – que les lecteurs de MISC connaissant mieux que l'auteur de ces lignes –, qui ne ressemblent

que très peu aux visions cinématographiques à la *Mission Impossible* ! À titre d'exemple, une action *low cost* de l'équipe de Mr X sera de :

- Établir par contact direct le type de matériel audio utilisé par NLIL lors des « grandes messes » (internes, investisseurs, etc.) sous le simple prétexte de proposer commercialement de nouveaux produits. À noter, généralement, ce matériel est « sans fil » pour simplifier le travail de l'orateur.
- Établir les dates de ces réunions par élicitation auprès des assistantes ad hoc. Un focus particulier sera fait sur les réunions internes de présentation des résultats, les réunions avec les investisseurs et analystes financiers, etc.
- Puis, disposer un scanner de type 300Mhz [17] en mode « *roam & search* ». L'écoute se fera à partir d'un poste fixe ou temporaire, avec un système d'enregistrement...

La cartographie des flux, blocages et « zones de stocks »

L'objectif des premières étapes du process est clairement de pouvoir disposer des informations ouvertes et de surveillance initiale pour déterminer les « angles d'attaques » ultérieurs ! Il existe une multiplicité d'approches concernant cette phase. Les plus universelles sont présentées ci-dessous.

Un rappel :

Les failles sont évidemment combinatoires, chacune pouvant s'appuyer sur les autres pour être plus efficaces ! Ainsi, on considère que le cœur est généralement la famille de failles opérationnelles, que les failles humaines renforcent toutes les autres, que les failles techniques et physiques facilitent les accès aux autres failles, etc.

Une multiplicité des approches

Les zones naturelles identifiées lors de la recherche de failles : après les phases initiales, l'équipe dispose de nombreuses informations facilitant la détection des « zones naturelles » exploitables. Quelques exemples : existence ou non d'un centre de reprographie, logiciels comptables utilisés, accès aux déchets, lieux de rencontres « hors travail » des cadres, horaires de travail, présence d'outsourcing (ménage, sécurité, etc. ;) sur les sites, etc.

La cartographie des flux à partir d'un modèle systémique : un des modèles les plus efficients lors de ce type d'opération est le modèle dit « de système-cible » qui analyse une entité en fonction de sa présence dans son environnement. La matrice permet de comprendre et de mettre en exergue les flux entrants et sortants, les « zones de stockage » de l'information générale ou sensible (archives, facturation, stockage physique, sauvegardes électroniques, etc.), les points d'entrée essentiels des canaux (un ex : l'assistante partagée par les commerciaux qui voit l'ensemble des résultats de prospection de ces derniers).

L'objectif des premières étapes du process est clairement de pouvoir disposer des informations ouvertes et de surveillance initiale pour déterminer les « angles d'attaques » ultérieurs !

La Multidimensionnal analysis (cf. Figure n°3) : l'approche est simple ! Elle repose sur l'intégration visuelle de 6 « unités basiques » (Espace = x, y, z / Temps = t / Énergie = H (enthalpy), S (entropie)) et permet – par exemple – parfaitement de comprendre les interactions entre un process de production et l'organigramme interne d'une société. Combinée au modèle systémique, l'analyse de cette matrice facilite la détection des flux vitaux et donc par la suite des « surfaces utiles ».

Le SNA (Social Network Analysis) : les approches par sociogrammes sont évidemment utiles pour comprendre les interactions entre les membres de l'entreprise. De nombreux modèles existent : allant d'outils universitaires disponibles en ligne à des outils « pointus » permettant de comprendre les interactions des emails entre individus d'une même entité. Fondamentalement, deux types de résultats sont attendus : premièrement, savoir « qui est proche de qui », et, ensuite, savoir « qui n'aime pas une source potentielle » (ce qui en fait de facto une excellente source d'information). Le SNA permet également de détecter les « isolés » de la structure, les points d'accès en contact avec l'extérieur (canaux d'entrée privilégiés), etc.

Les objectifs

Cette phase a plusieurs objectifs somme toute assez simples :

1. Identifier les flux importants par rapport aux objectifs de collecte ;
2. Identifier les points d'accès les plus logiques ;
3. Pré-identifier le type d'individus pouvant avoir accès aux informations recherchées.

La sélection des surfaces utiles et l'analyse de risques

Une fois la cartographie réalisée, les opérateurs vont travailler sur la sélection des « surfaces utiles ». Quid d'une surface utile ? Il s'agit de la zone au sein des canaux/flux qui a la plus forte probabilité de « contenir » les informations recherchées. Évidemment, il n'existe pas une seule surface utile, mais plutôt des surfaces utiles en fonction de la typologie des informations recherchées.

Ces surfaces déterminées, les opérateurs vont appliquer une matrice de risques qui permettra de catégoriser les surfaces en fonction d'un ratio « qualité des infos/prise de risques » pour l'équipe. Rappelons, que l'axiome numéro 1 du métier de prédateur est « ne pas se faire prendre » !

L'analyse de risque consiste fondamentalement à faire le travail inverse d'un service de sécurité en soulignant :

- l'importance des mesures de protection des données ;
- le risque de raté de l'opération d'accès ;
- le degré d'analyse des incidents par la sécurité ;
- le niveau des OPSEC à prévoir ;
- La logistique, etc.

Cette étape comprend également une analyse de la qualité potentielle de l'information avec des critères comme le degré d'exploitation possible des données (cryptées ou en clair, shreddées ou non, etc.) ou le degré de proximité avec la surface la plus productrice de l'information (proche ou non du service comptable et du DAF pour le financier, de la stratégie pour les plans de développement, etc.).

L'identification d'individus précis



4 *Les grandes catégories d'internes exploitables dans l'entité (classé selon la dangerosité (risque/capacité de contrôle sécuritaire)*

Si les étapes préalables ont été correctement menées, l'équipe de Mr X devrait avoir une vision assez précise des individus qu'elle devra cibler.

En général, **les opérationnels disposent de leurs propres grilles d'analyse qui reprend les différentes catégories d'individus les plus intéressants dans une société**. Quelques exemples : les « invisibles » ayant accès aux informations sensibles (employé du centre de reprographie, femmes de ménages, assistantes, etc.), cadres en charges de projets sensibles, comptables, direction commerciale, etc.

Ces catégories standards sont ensuite analysées en fonction d'une grille des profils stéréotypés les plus fragiles (cf. Figure n°4). Sont recherchés à ce stade : la frustration individuelle ou d'un groupe (ex de sources : les syndicats, les discussions au restaurant ou mieux devant une bière le soir, etc.), les anciens employés (si possible en conflit ouvert ou larvé), les employés sur le départ (avec analyse des motivations : rappelons que ces employés ont encore leurs accès et qu'ils peuvent être « fragiles » sur des approches en faux recrutements, des rancœurs personnelles, etc.), évidemment les extérieurs présents in situ (sociétés de services de secrétariat, d'accueil, femmes de ménages, maintenance informatique, sécurité, etc.).

L'équipe s'attachera à « mettre des noms » en face de chacune de ces catégories en utilisant les mêmes approches combinées de surveillance/collecte que dans la phase initiale.

Un simple exemple concernant NLIL : les restaurants dans les zones industrielles aux USA ont généralement des tirages au sort de *freemeal* hebdomadaires. La règle est simple : mettre une carte de visite business dans le petit panier à côté du comptoir ! Un opérateur peu ainsi facilement lors du « coup de feu » analyser les CV dans le panier et identifier des noms, des fonctions, et des « structures informationnelles » exploitables par la suite (structure des emails, code courrier interne, logique des lignes téléphoniques, etc.).

Une fois cette liste disponible, une analyse plus fine des failles humaines est initiée pour sélectionner les « forts potentiels ». Toutes les grilles d'analyse sont alors déployées : fragilités basiques (MICE), émotionnelles, stéréotypes de profils à risques (cf. Figure n°5), psychologique et de personnalités, fragilités médicales (en général obtenues par accès aux WC et SdB personnels de la source sous un prétexte quelconque).

Cette analyse s'appuie cette fois-ci sur une surveillance approfondie et ciblée [18] sur les individus sélectionnés ! Juste un point ici : il ne s'agit pas de connaître le coût de ces opérations, mais bel et bien de mettre autant de moyens techniques, humains, financiers que possible sur cette phase

La personne immature : cible facile pour un professionnel. Risque augmenté lorsque l'immaturité se combine au besoin « d'appartenir à ceux qui savent »

Le « héros » : ou plus précisément le « héros révélé » qui se voit offrir de changer la société alors que jusqu'alors sa vie était terne et banale...

L'amateur d'intrigue : attiré dans ce monde par son seul goût naturel...

L'insatisfait : qui n'accepte pas de vivre une vie médiocre ! D'où une recherche perpétuelle de sensation capable de flatter son ego. Au-delà des produits, nos entreprises actuelles « fabriquent » des masses d'insatisfaits fragiles d'autant plus en période difficile (réduction de personnels, etc.)

Le solitaire : dans une société de communicants, les personnes souffrant de solitude se retrouve marginalisées dans les sociogrammes. Ils sont une proie facile! Les SR soviétiques avaient ainsi monté une opération dédiée (Myosotis) auprès d'occidentaux ... Les arnaqueurs russes font de même aujourd'hui avec les sites de rencontres !

L'intellectuel : qui de part son intelligence, a un besoin permanent d'évoluer. Si cette évolution est limitée et/ou lente : il ressent une frustration. Cette frustration se reporte généralement sur l'environnement (pays, société, famille, etc.)

5 *Quelques stéréotypes d'individus « à risque » recherchés par les professionnels pour leurs fragilités*

qui « assure » la réussite des recrutements ultérieurs ! L'objectif est « de savoir ce qu'il faut savoir » pour manipuler l'individu lors du recrutement. En particulier, ce que l'individu ne veut pas que les autres sachent de lui... MrX peut ensuite passer à la phase suivante : le recrutement de sources humaines.

Dans le cas de NLIL, Mr X se focalisera essentiellement sur 5 individus :

- **Brian – cadre supérieur de NLIL, frustré et en colère parce qu'il estime qu'une promotion lui est « passé sous le nez »** : qui « déballe » sa rancœur à toute personne voulant bien l'écouter dans son bar favori ! L'approche s'appuiera sur une approche d'amplification de son ego, doublée d'excuses pour expliquer ses actes futurs.
- **Angie – hispano-américaine, ayant un jeune fils proche d'un street-gang du Downtown** : dont le nom est sorti lors d'une recherche sur les *county courts* locales avec une comparution du fils dans les semaines à venir.
- **Bob – livreur de fontaine d'eau et « Mister Sandwich », croulant sous les dettes de ses crédits revolving, personnalité asociale** : LE modèle idéal pour un recrutement ouvert : l'asocial n'ayant pas de loyauté ; adorant les prises de risques (jeux risqués, sexe, etc.) et, plus important encore, cherchant des moyens simples de résoudre ses problèmes !
- **Ed – comptable de 46 ans, marié, amateur de travestis à ses heures perdues** : la solution du *blackmail* n'est pas retenue compte tenu de son risque comme approche initiale. Mr X utilisera un *service agent* [19] – prostitué homme de son état – pour ce recrutement.
- **Pat – 36 ans, seul employé du centre de reprographie de NLIL, solitaire, sans charisme** : seul depuis plusieurs années, ne pouvant recourir moralement à des prostitués, et évangéliste.

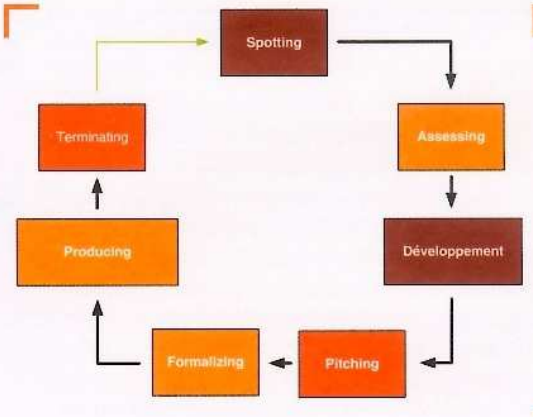
Contrairement à une idée répandue, sur quelques centaines d'employés, il est relativement facile de trouver quelques dizaines de « potentiels », et donc quelques sources de ce type !

L'Agent Acquisition Process (Figure n°6)

Puisque notre exemple a pour cadre le sol américain, nous utiliserons la terminologie en vigueur dans les services spécialisés US nommé l'*Agent Acquisition Process* ou l'*Asset Acquisition Process* (le choix du terme « asset » étant en lui-même révélateur d'un certain état d'esprit).

Ce process comprend 7 phases distinctes. Dans la figure 6, le lecteur attentif aura remarqué un choix de couleurs pour ces différentes étapes. Il s'agit d'une représentation graphique des risques inhérents pour l'opérateur des différentes phases : jaune = peur de risque d'être découvert ; brun = risque d'identification des opérateurs nécessitant la mise en place de procédure d'OPSEC spécifique ; orange = phases au cours desquelles le risque est le plus élevé pour l'opérateur.

Avant d'aborder le process à proprement parler, il convient de noter que derrière chacune des phases, l'opérateur dispose d'un ensemble de méthodes et « outils » lui permettant de réaliser son recrutement dans les meilleures conditions. Ces méthodes vont au-delà du cadre de cet article et, d'ailleurs, ne sont pas utiles pour mettre en place une défense efficace.



6 *L'Agent Acquisition Process*

Rappelons également que malgré la rigueur de la méthode, un recrutement reste un pari – certes peu risqué – pour l'opérateur : un être humain ayant parfois des réactions peu ou pas prévisibles lorsqu'un *pitch* lui est proposé...

Décomposons maintenant nos 7 phases :

1▶ **Spotting** : cette phase reprend la partie d'identification d'un individu précis. Généralement, elle fait l'objet d'approches discrètes du sujet (élicitation par exemple) pour valider l'intérêt de la source potentielle au-delà de ses fonctions « prometteuses ».

2▶ **Assessing** : phase plus ou moins longue (de quelques semaines à quelques mois en environnement protégé), les objectifs sont simples : connaître l'individu, le profiler **[20]**, identifier les failles les plus exploitables et les moins dangereuses pour l'opérateur. C'est ici qu'intervient le métier du traitant ! Rappelons que le process est discret : la source ne devant pas se rendre compte de cette phase ou des suivantes jusqu'au pitch.

3▶ **Development** : la phase de développement de la source est complexe, délicate et potentiellement dangereuse pour l'opérateur. L'objectif est de « faire partie » de l'environnement de la source, puis de développer une relation – de plus en plus intime – avec cette dernière. En clair : « devenir le nouvel meilleur ami » de sa source ! Selon le réseau initial retenu (professionnel, privé, etc.), et les approches sélectionnées, cette phase est plus ou moins longue. C'est également lors de cette phase que l'*assessing* est poursuivi cette fois-ci pour valider les failles identifiées. Ainsi, le traitant peut proposer dans une escalade parfaitement maîtrisée, à la source de « s'investir » de plus en plus dans la relation tout en testant ses failles. Un simple exemple : un opérationnel ne demandera pas de but en blanc une copie de l'annuaire interne de NLIL ! L'approche retenue est liée à une logique d'accoutumance. Il commencera par rendre service à la source, puis à demander un simple numéro de téléphone interne. Puis, plus tard, un nom précis. D'un nom, il ira vers la présentation d'une « histoire » permettant de tester certaines failles utiles à ce stade (MICE/ASIE), par exemple, qu'un ami à lui – recruteur chez un chasseur de tête – recherche des acheteurs spécialisés dans l'achat de confection textile en Asie. Il demandera un service à la source pour identifier au sein de NLIL des « cadres à potentiels ». Si la source accepte, il poursuivra son recrutement. Si non, il prévoira une « sortie » et testera un peu plus tard à nouveau la source. Puis, plus tard, il expliquera à la source que son ami consultant était ravi des informations, et qu'il veut rétribuer la source pour le service. Une enveloppe sera remise : encore un test ! Enfin, le même consultant demandera une copie de l'annuaire interne. Le traitant offrira dans le même temps des « excuses » à sa source afin d'alimenter la rationalisation de cette dernière **[21]**. Une fois amené à un point précis permettant le pitch, l'opérateur pourra soit réaliser ce pitch seul, soit « transférer » la source au pitcher. L'intérêt de transférer le contact est de renforcer les OPSEC en cas de problèmes futurs et également de ne pas remettre en cause directement la relation source/opérateur1 en cas d'échec du pitch.

4▶ **Pitching** : une fois « mûre », les opérationnels proposent à la source un pitch ! En clair, il s'agit d'une offre claire de recrutement pour une opération de renseignement. Évidemment, si l'offre est « claire », la présentation de cette dernière peut faire l'objet de prétextes particuliers. L'équipe travaillera plusieurs jours pour affiner l'histoire servie à la source afin de limiter les risques de refus ! C'est ici que les résultats des phases précédentes trouvent tous leurs sens... Plus la source est « connue », plus il est possible de « customiser » l'offre. L'opérateur2 utilisera tous les moyens possibles pour faciliter ce passage. Ainsi, dans certains cas de figure une opération sous « faux drapeau » sera montée. Quid ? C'est simplement une légende plus complexe destinée à protéger l'opérateur en cas d'échec et à augmenter le taux de réussite. Un exemple du milieu du renseignement : dans les années 70, un couple d'israélites britanniques – dont le mari était au Foreign Office – transmettent de l'information classifiée, à un traitant d'un pays arabe – se faisant passer pour un membre des services israéliens – sous le prétexte que le FO privilégie les pays arabes au détriment d'Israël ! Évidemment, le traitant agit comme un officier des SR israéliens, amenant même le couple à accepter une immigration pour « services rendus à la patrie » après leurs missions... **[22]** De l'argent est également versé pour « soutenir » la famille de ce jeune père vivant chichement pour mieux enferrer la source. Comme le soulignait Victor Ostrovsky dans son ouvrage le plus connu : « *The idea of recruitment is like rolling a rock down a hill...You take somebody and get him gradually to do something illegal or immoral. You push him down the hill. We didn't blackmail people. We didn't have to. We manipulated them* ».

5▶ Formalizing et 6▶ Producing : il s'agit ici des phases les plus simples pour l'opérateur. La première consiste à fixer le cadre de la collaboration et la rémunération. La seconde de mettre en place les transmissions de besoins et en retour d'informations au moyen de techniques spécifiques à ce métier. Évidemment, l'opérateur continue d'évaluer en permanence la source – mais aussi les informations transmises – et peut régulièrement avoir à mettre un peu de pression sur cette dernière en cas de réticences. Il offre également des « moyens de résoudre » la dissonance cognitive permanente ou ponctuelle de la source. Dans le cas de NLIL, Mr X formera également certaines de ses sources à des méthodes efficaces de collectes illégales. Ainsi, Brian se verra former à utiliser des enregistreurs logiciels sur PDA qui donnent l'illusion que le PDA est éteint **[23]** pour enregistrer les réunions intra ou interservices (une analyse de ces données permet 1- de vérifier la fiabilité de la source, 2- d'avoir un meilleur débriefing, 3- d'analyser le paraverbal des individus présents (motivations, degré de tension, liens entre les individus, etc.), mais également à remplir des questionnaires précis sur ses collègues lors des réunions (objectif de détection de failles, *profiling*) ; Angie, elle, sera formée à l'utilisation de clés USB contenant des Troyens, à réaliser des *drafts* des bureaux (y compris objets personnels pouvant servir de camouflage à des microphones), à la pose d'outils de sonorisation dans les lieux ad hoc, à lire les *rolodex* des cadres identifiés ; Bob notre homme « porteur d'eau » matinal, aux techniques dites de l'éléphant **[24]** qui permettront petit à petit d'obtenir des passwords des individus dans les bureaux situés à côté des fontaines d'eau ; Pat, notre employé du centre de reprographie sera initié aux méthodes d'aspiration des disques durs des copieurs ; Bob à la copie discrète de fichiers informatiques, et à la reprographie des documents protégés non copiables, etc. Ces formations se déroulant généralement dans une *safe house* **[25]** dont le cadre reproduit les conditions réelles. Pourquoi former des sources ? Pour assurer les OPSEC de l'opération premièrement **[26]**, et assurer une meilleure productivité en termes de collecte tout en les « protégeant » le plus possible pour éviter leur identification. Enfin, une formation même succincte aux méthodologies du renseignement donne l'illusion aux sources d'être « dans le secret », et calme les plus anxieux.

encadré 1

Les résultats en termes d'informations obtenues, envisageables avec ces recrutements sont de type :

- copie partielle ou intégrale du book comptable > accès à l'ensemble de la structure de coûts et de rentabilité de NLIL ;
- compréhension du système de sous-traitance de NLIL y compris les *sweetshops* illégaux utilisés en Asie ou ailleurs ;
- compréhension du service commercial et de son mode de fonctionnement ;
- possibilité de pénétrer – ou pénétration – dans le réseau informatique à partir des passwords, etc.
- sonorisation des lieux importants : bureaux du CEO et du CFO, salle de réunion, toilettes dame ; toilettes des cadres, etc.
- organigramme complet y compris informations de rémunération ;
- dossier d'environnement sur l'ensemble des cadres importants de NLIL – y compris failles, possibilité de compromission, etc. - ;
- listes des clients, fournisseurs, prospects ;
- copies des documents passant sur les imprimantes ;
- renseignement d'ambiance sur les difficultés, les forces, les faiblesses de NLIL ;
- renseignement d'ambiance sur les liens entre individus et les sociogrammes ;
- etc.

7▶ Terminating : toute bonne chose ayant une fin, il arrivera à un certain stade que la source devienne inutile. Plusieurs raisons expliquent cet état de fait : la source peut être mutée et n'a plus d'accès intéressants ; le projet est terminé, elle est donc inutile ; sa qualité est médiocre, etc. Dans tous les cas de figure, la situation est dangereuse pour l'opérateur ! En effet, la source « lâchée » peut ressentir de l'animosité, un manque (relation privilégiée), une envie de se dénoncer au service de sécurité, un manque financier (la source de revenus se tarissant), etc. Dans tous les cas de figure, cette phase nécessite un talent certain de l'opérateur pour « terminer » la relation dans les meilleures conditions de sécurité ! Dans le monde du renseignement industriel, cette phase peut être accompagnée d'un bonus, et aller jusqu'à l'exposé de mesures de rétorsions envers la source qui voudrait se dénoncer...

Afin de poursuivre notre exemple sur NLIL, voici quelques exemples concernant certains de nos individus :

Ces exemples sont nécessairement simplifiés compte tenu du format de l'article, chacun nécessiterait plusieurs pages déroulant la méthodologie précise. Néanmoins, l'auteur les espère suffisamment explicites pour comprendre le déroulé de la méthodologie !

⇨ **Brian** : est accroché dans son bar favori (identifié par surveillance physique), Mr X « écoute » sa rancœur. Il donne l'impression à Brian que c'est ce dernier qui initie la conversation (technique basique utilisée dans toutes les approches pour limiter la méfiance de la source). Après un échange de CV, un déjeuner, quelques dîners que Mr X prend à sa charge. Une légende simple : Mr X est un broker financier... Puis, petit à petit, le broker révèle qu'il aime faire des coups « limites » en disposant d'informations sensibles sur les sociétés. Il entraîne Brian lentement d'un simple service à un recrutement, évidemment rémunéré ! Tout en « soutenant » l'ego de Brian et en lui offrant des justifications pour son comportement et le fait « qu'il doit être pris au sérieux et rémunéré en conséquence ». Brian adore également les *tradecrafts* enseignés par Mr X qui lui permettent de « faire partie de ceux qui savent » et de se moquer intérieurement de ses « stupides collègues » !

⇨ **Bob** : personnalité asociale typique, Bob prend tous les risques ! Ainsi, tout à sa passion des tables de poker, il aime rejoindre des « amis d'amis » pour une petite partie improvisée. Un vendredi soir après quelques bières, Jack une vieille connaissance, lui parle de Ted – un joueur qui joue gros, mais joue mal – et qui cherche une table régulière. Bob comprend de suite, l'intérêt de rencontrer ce Ted et demande des informations sur les endroits fréquentés par ce dernier. Quelques jours plus tard, il rencontre Ted et lui propose de se joindre à une petite table entre amis. Ce soir là Bob gagne et plume Ted ! Heureusement, cela lui permet de limiter la pression de ses créanciers ! Bob et Ted sympathisent. Ted écoute les « malheurs » de Bob et lui propose de rencontrer un ami à lui qui l'a déjà aidé en situation difficile en échange de quelques « menus services ». Bob rencontre Mr X qui se présente comme un PI qui cherche des informations pour « coincer » des hommes ou femmes mariés en démontrant leur infidélité (fortement punissable en Californie). À ce titre, Mr X explique à Bob que ses visites régulières dans toutes les sociétés du coin sont une bénédiction qu'il est prêt à rémunérer en cash si Bob sait ouvrir les yeux... En demandant la liste des sociétés, Mr X apprend avec surprise que NLIL où il a une cible, fait partie de cette dernière ! Il propose à Bob de tester leur association sur NLIL ! Bob sera ainsi utilisé sur NLIL, mais évidemment, sera également un excellent service agent ou *access agent* dans les autres sociétés locales.

⇨ **Ed – comptable de 46 ans, marié, amateur de travestis à ses heures perdues** : la solution du blackmail n'est pas retenue compte tenu de son risque comme approche initiale. Mr X utilisera un service agent [27] – prostitué homme de son état – pour ce recrutement. Une fois déterminés les lieux de rencontres de Ed pour ses aventures, Mr X demande à son service agent – Alex – d'établir le contact ! Alex évidemment ne se présente pas comme prostitué, mais comme un jeune homme fragile, qui se sent femme... Sans expérience et à la recherche du « grand amour ». Les relations sexuelles avec Ed sont filmées et sonorisées. Mais l'objectif réel est de « faire fondre le cœur » d'Ed en ayant une relation plus amoureuse que simplement sexuelle. Au bout de plusieurs semaines, cette relation qui prend corps entraîne Ed dans une double valence complexe entre son envie de rester avec Alex, et sa vie « normale ». L'objectif est de créer une détresse psychologique qui restera nécessairement cachée pour sa femme. Alex devient alors, l'amante, l'amoureuse, la confidente d'Ed. Plus le lien est fort, plus Alex s'ouvre sur ses vrais désirs, comme subir une intervention faisant de lui, une vraie femme ! Alex et Ed – qui approuve cette idée – se renseignent sur le coût de cette opération qui reste hors de prix ! Déçu, Alex sombre dans une « dépression ». Ed essaye de protéger ce dernier. Puis, un jour, Alex explique qu'il a une solution simple : il connaît un type qui peut les payer pour quelques informations ! Ed refuse, puis finit par plier devant un subtil mélange de menace de suicide, sexe, pleurs, qu'Alex a répété avec l'équipe de Mr X. Ed rencontre Mr X, Brad en l'occurrence qui sous couvert de son travail comme *Competitive Intelligence Officer* pour un capital-risqueur – qui envisage d'investir dans NLIL – lui demande d'obtenir des copies de certains documents comptables en échange d'une somme d'argent permettant à Alex de devenir une femme...

⇨ **Angie** : lors de la comparution initiale de son fils, rencontre Helena, une jeune employée d'un *advocacy group* spécialisé dans le soutien des populations à revenus limités dans le comté de Los Angeles. Cette dernière propose à Angie de lui offrir les services d'un bon avocat pour défendre son fils et éviter ainsi l'avocat de l'aide judiciaire. Quelques semaines plus tard, grâce à l'action de l'avocat, le fils obtient une peine réduite pour ses actions et un sursis. Plus tard encore, Helena explique à Angie que son advocacy group est révolté par les conditions de travail des Chicanos dans les usines textiles US et demande à Angie son aide pour mieux comprendre les pratiques de ce secteur ! Reprenant le principe de l'escalade, Angie est petit à petit amenée à réaliser des actions illégales pour le compte d'Helena. Cette dernière a évidemment assorti sa demande d'aide enthousiaste, de menaces à peine voilées sur la fragilité du sursis de son fils sans l'aide de l'advocacy group... Cette association fait évidemment partie de la légende d'Helena – qui ne s'appelle pas Helena –, et qui fait partie de l'équipe de Mr X.

Pat : à part ses offices religieux, Pat a une vie terne et solitaire ! Réveil à la même heure, transports en commun jusqu'au travail. Repas. Re-travail d'un ennui mortel (la reprographie n'est pas la plus excitante des fonctions !). Re-transport. Re-repas, mais en solitaire cette fois-ci. Un film, parfois pornographique (ce que Mr X sait par la liste des films loués par Pat au DVDLocation du coin sous un faux prétexte). Bref, une vie terne… D'autant plus que Pat n'a pas vraiment d'amis et, est plutôt banal physiquement. Jusqu'au moment où tous les matins, une belle femme se retrouve systématiquement au même arrêt de bus. Pat aimerait bien l'aborder, mais n'ose pas ! Cette fille est d'une classe au-dessus de la sienne… Elle, de temps en temps, lui dit bonjour. Lui sourit. Pat comprend qu'en fait, Sylvia ne prend pas le bus du matin. Elle se met sous l'abri le temps qu'un superbe Hummer flambant neuf avec un mec « très sûr de lui » passe la chercher 5mn avant son propre bus. Un matin pourtant, Sylvia n'a pas le sourire, elle ne cesse de regarder sa montre ! Son ami – et sûrement amant pour Pat – est en retard. Excédée, elle prend le bus. 2mn plus tard, le Hummer apparaît et l'homme – Franck – demande à Pat s'il a vu Sylvia ? Quand il apprend qu'elle est montée dans le bus, Franck est contrarié… Mais propose quand même à Pat de l'emmener une fois qu'il connaît sa destination qui par hasard est dans le même direction ! Chaque matin ensuite, Franck prend Sylvia et Pat dans son Hummer. Pat apprend incidemment que Sylvia n'est pas la petite amie de Franck. Diana – « petite amie » de Franck – et ce dernier invitent un soir Pat et Sylvia, puis plusieurs fois en s'occupant des factures. Pat angoisse de ne pas « être à la hauteur » ! Ni intellectuellement, ni financièrement. Puis, Sylvia couche avec Pat. Un couple se forme sous la bienveillance de Franck qui explique à Pat qu'il est heureux qu'il soit « normal » pas comme les ex habituels de Sylvia qui sont des gamins arrogants et pourris par l'argent. Petit à petit, sur une escalade, Sylvia commence à coûter cher à Pat qui veut évidemment « la sortir seule ». Service agent (et prostituée) de son état, Sylvia applique ainsi le plan de Mr X – Franck pour les intimes – qui consiste à plonger Pat dans la détresse psychologique, et la pression financière pour « garder » Sylvia après ces années de solitudes ! Pat est ensuite « récupéré » par Franck qui lui offre « par amitié » une possibilité de gagner un peu d'argent pour payer à Sylvia son week-end de luxe à Hawaï dont elle « rêve tant pour son anniversaire ». Pat est ferré, Mr X n'a plus qu'à amplifier peu à peu le mix entre pressions et aides… Et à former Pat.

Évidemment, il s'agit d'un *case study* quasi parfait en termes de recrutement ! Chacun étant réalisé sous un « faux drapeau » différent avec un cloisonnement parfait ! Néanmoins, les actions présentées sont inspirées de cas réels…



Conclusion

Si l'exemple de NLIL est extrêmement simplifié, il n'en reste pas moins proche des procédés actuellement utilisés par l'ensemble des prédateurs informationnels lorsqu'ils jettent leurs dévolus sur une cible ! Dans notre exemple, il faudrait probablement environ 6 mois de la phase initiale à l'analyse de la pleine « production » des sources humaines.

Rappelons un axiome de tous les métiers collectant de l'information : « l'information est au centre des process de l'entreprise » ! Nous rajouterons que sans informations fiables, aucun prédateur n'est capable dans sa spécialité de « toucher » sa cible ! Qu'il s'agisse de collecte, de désinformer, de perturber, de voler ou encore d'arnaquer. Le renseignement est la fondation sur laquelle le prédateur s'appuie pour monter ses opérations ! Et, dans ce cadre, l'identification des failles et des individus est LE process essentiel qui détermine l'efficience de toutes les actions ultérieures…

L'auteur rappelle, par ailleurs, qu'il condamne fermement le recours à ces approches offensives – souvent illégales – sauf dans le cadre fixé contractuellement et légalement, d'un test de pénétration informationnelle. D'ailleurs – y compris dans ce cas –, il est illégal et d'ailleurs non éthique de monter de vraies opérations de recrutement d'employés ! Il est cependant obligé de convenir que les Spooke **[28]** ne sont pas rares dans les événements professionnels comme les salons… Ce qui souligne probablement une forme de demande plus ou moins discrète.

Juste pour « briser le cou » à un vieux fantôme : dans le contexte actuel, toute entreprise peut devenir la cible pour de multiples raisons de ce type de prédateurs ! Plus évidemment, des opérateurs du renseignement industriel et étatique sont impliqués, plus la cible a peu de chance de détecter les actions compte tenu du degré de professionnalisme de ces acteurs. Mais rappelons quand même qu'un simple arnaqueur avec des méthodes basiques est capable de vous coûter très cher en temps et en argent !

Fort heureusement, en comprenant les modes opératoires des prédateurs informationnels, il est possible de mettre en place des procédures de protection. Ces dernières ne peuvent être exposées ici, mais feront éventuellement l'objet d'un article futur.

Notes

[1] À titre d'illustration, le document *Russian Business Network Study* de David Bizeul, est une étude intéressante sur cette catégorie de prédateurs : **www.bizeul.org/files/RBN_study.pdf**.

[2] Membre d'un service de renseignement en charge du traitement des sources d'origine ROHUM (renseignement d'origine humaine).

[3] Confère une présentation de l'auteur sur les malveillances et l'impact sur les entreprises réalisée en novembre dernier auprès de l'auditoire de l'IERSE.

[4] OPSEC : Sécurité opérationnelle des opérations de renseignement offensif. Les OPSEC comprennent tous les outils et méthodologies permettant d'assurer à un opérateur qu'il ne sera pas découvert et/ou permettant de diriger la sécurité adverse vers de fausses pistes.

[5] Évidemment, Alpha grimacera quant au montant de la facture ! Ce type d'opération est en effet très coûteux…

[6] Il existe une pléthore d'ouvrages sur l'acquisition des données ouvertes à grises aux USA : par exemple les ouvrages de la société Washington Researchers, et de nombreux brokers spécialisés par typologie de documentation sont répertoriés allant du plus sérieux au plus illégal ! (**www.washingtonresearchers.com**).

[7] En toute logique et pour respecter la logique des OPSEC, il est probable qu'une grande partie de ces actions soit réalisée en sous-traitance par des PI (*Private Investigator*) locaux. À titre indicatif, l'usage des détectives est nettement plus réglementé en Europe qu'aux USA.

[8] Société « coquille vide » utilisée pour disposer de prétexte à la collecte (les plus courantes : éditions, consulting, presse, logisticien, chasseur de tête, etc.)

[9] Un cas célèbre a défrayé la chronique lors de la seconde Guerre du Golfe. Un journaliste US a ainsi utilisé l'augmentation de la fréquence et volume des commandes du Pentagone chez Domino Pizza pour identifier la « fenêtre de tir » de l'attaque US…

[10] IWOCHEWITSCH (Michel), « Les failles humaines et l'information », *MISC* 34, nov.-déc. 2007.

[11] **www.healthbolt.net/2007/02/14/26-reasons-what-you-think-is-right-is-wrong/**

[12] IWOCHEWITSCH (Michel), « Les failles humaines et l'information », *MISC* 34, nov.-déc. 2007.

[13] Ibid.

[14] Mise sous surveillance par microphone.

[15] Selon des rumeurs fréquentes ces dernières années, des sociétés de la City ont fait l'objet de chantage à l'EMP exploitée pour détruire les supports physiques des transactions financières de la journée.

[16] Un « bon » document sur le phénomène Tempest : **http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-577.pdf**

[17] Le coût est faible et l'acquisition aisée sur Internet…

[18] Pouvant inclure sans être limitatif, le *waste archeology*, la surveillance physique, la sonorisation des lieux de vie, le piratage des PC privés, l'approche d'individus dans divers réseaux de la source, etc.

[19] Les opérateurs disposent en permanence de plusieurs agents de ce type recrutés par ailleurs et qui peuvent servir sur plusieurs opérations comme le service agent (apportant un support tel que le sexe, la drogue, pouvant réaliser des surveillances, etc.) et l'access agent (ayant un accès direct à une source ou à un lieu).

[20] Confère l'article de l'auteur dans *MISC* 34.

[21] Phénomène de dissonance cognitive décrit dans l'article *MISC* sur les failles humaines.

[22] Pour l'histoire, le mari n'a toujours pas retrouvé la liberté. Le couple est évidemnent *persona non grata* en Israël compte tenu des dommages causés au pays.

[23] De nombreux outils de ce type sont disponibles sur Internet.

[24] Nommé ainsi par les services de sécurité canadiens selon la vieille blague « comment mange-t-on un éléphant ? Morceau par morceau… »

[25] « Maison sûre » louée généralement par un service agent pour réaliser spécifiquement les formations/débriefings en respectant les OPSEC. Dans les cas les plus dangereux, les opérateurs auront différentes safe house pour les sources afin d'augmenter le cloisonnement et diminuer la prise de risque.

[26] À ce titre, les sources seront également formées aux méthodes sécurisées de transmission des informations physiques (BLM, lieu physique de dépôt des documents en « boîte lettre morte » avec un système de signalisation et une protection) ou électroniques selon des procédures rigoureuses protégeant les opérateurs.

[27] Les opérateurs disposent en permanence de plusieurs agents de ce type recrutés par ailleurs et qui peuvent servir sur plusieurs opérations comme le service agent (apportant un support tel que le sexe, la drogue, pouvant réaliser des surveillances, etc.) et l'access agent (ayant un accès direct à une source ou à un lieu).

[28] Un des « petits noms » des experts privés du renseignement industriel.

Benjamin Caillat – bcaillat@security-labs.org

OUTIL SPÉCIFIQUE POUR ATTAQUES CIBLÉES D'ENTREPRISES (PARTIE 1)

Les attaques informatiques ne sont aujourd'hui plus le seul fait de hackers passionnés développant des codes pour la beauté de l'art ou pour se faire une réputation auprès de leurs pairs. Elles sont désormais également le fait de bandes organisées menant ces actions offensives dans un seul but : gagner de l'argent.

mots clés : *développement / backdoor / attaques ciblées d'entreprises / canal caché / furtivité / cryptographie*

1. Introduction

Dès lors, différentes entités peuvent être ciblées. Les ordinateurs des particuliers par exemple ; l'objectif est dans ce cas de voler les informations bancaires des utilisateurs ou de prendre le contrôle de la machine afin d'exploiter ses ressources pour mener des DDoS ou envoyer du spam.

Ces attaques visant les particuliers se déroulent dans un contexte très spécifique : les cibles sont innombrables, les objectifs relativement standardisés et le gain financier pour chaque piratage reste très modeste. Elles reposent par conséquent sur une automatisation et sur la loi du nombre. Les outils d'attaque employés combinent ainsi une propagation automatique (vers, *mass-mailing*, etc.) et une charge effectuant des opérations standards : intégration de la machine dans un réseau de zombies, collecte d'informations personnelles monnayables, chiffrement de fichiers en vue de chantage, etc.

Les systèmes d'information des entreprises peuvent également être ciblés. L'attaque vise alors les ressources critiques de l'entreprise : données confidentielles, flux financiers, disponibilité d'un service, etc.

Le contexte de l'attaque est alors bien différent de celui évoqué précédemment. Tout d'abord, du fait de la complexité de l'objectif et des spécificités du système d'information, l'attaque ne peut généralement pas être automatisée. Ensuite, les risques encourus par l'attaquant sont bien plus élevés : les systèmes informatiques sont parfois bien protégés et l'entreprise possède suffisamment de ressources pour faire appel à une équipe d'analystes post-mortem et mener une action en justice. L'échec doit donc être à tout prix évité. Quels outils sont donc adaptés pour mener une telle attaque ?

Cet article se propose de donner un bref aperçu des caractéristiques d'un outil utilisé pour mener des attaques ciblées d'entreprises. Il se place pour cela sous l'angle d'un attaquant choisissant de recréer complètement un tel outil. Il aborde la problématique par une rapide analyse fonctionnelle avant de présenter l'architecture logicielle qui en découle.

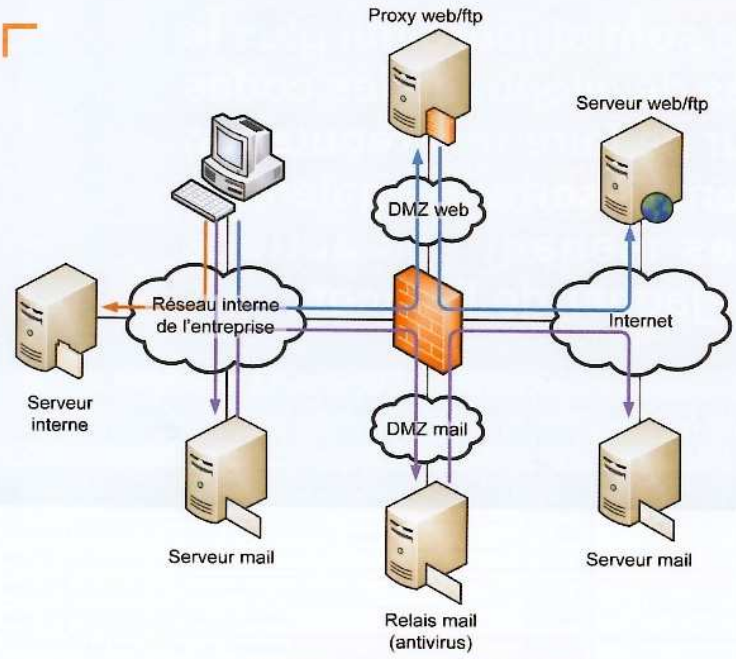
Ces résultats reposent essentiellement sur des développements personnels. L'objectif n'est pas de tenter de dresser une liste exhaustive de possibilités, mais de présenter les choix techniques effectués lors de ce développement.

Une partie des techniques présentées dans ce document ont été implémentées et validées sur des architectures de tests. Une version bridée de l'outil développé et quelques ressources présentant ces résultats sont disponibles [1].

2. Définition du contexte

2.1 Objectif et cible

Les objectifs de l'attaque peuvent être très variés : récupération d'informations confidentielles, espionnage d'un employé, modification de valeurs métier dans le système d'information, etc.



1 *Modélisation de l'architecture réseau de l'entreprise*

Pour prendre un exemple précis, nous considérons une attaque ciblée visant à récupérer des informations confidentielles sur un réseau d'entreprise « standard » (cf. paragraphe suivant), tout en gardant à l'esprit que l'outil doit être suffisamment souple pour s'adapter à différents contextes.

2.2 Architecture réseau considérée

Nous modéliserons le réseau de l'entreprise par l'architecture schématisée sur la figure 1, qui reprend les éléments fondamentaux des infrastructures généralement mises en place.

Cette architecture est bâtie autour d'un *firewall* séparant la zone « externe », deux DMZ et la zone « interne ». Aucune communication directe entre les zones « interne » et « externe » n'est autorisée. Tout échange de données entre ces zones doit passer par les serveurs relais dans les DMZ.

Les protocoles classiquement autorisés entre un client du réseau interne et un serveur sur Internet via un relais sont : HTTP(S), SMTP, DNS et éventuellement FTP.

Le réseau interne est composé de postes et de serveurs Windows. Il faut noter que, selon l'état de l'art, il devrait être segmenté en plusieurs parties (physiquement ou via des VLAN) séparant les machines en fonction de leur rôle (comptabilité, développement, administration, etc.).

Les postes de travail sont équipés d'un antivirus à jour et les utilisateurs sont logués dans une session « utilisateur restreint » et n'ont donc pas les privilèges « administrateur ».

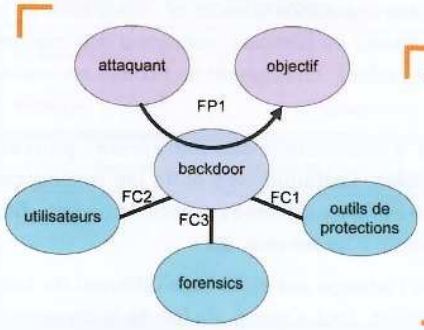
3. Analyse fonctionnelle de l'outil

La fonction de l'outil est d'aider l'attaquant à atteindre son objectif sur le réseau interne de l'entreprise. Dans la mesure où il offre un accès dérobé au système, il sera désigné dans la suite par le terme « *backdoor* ».

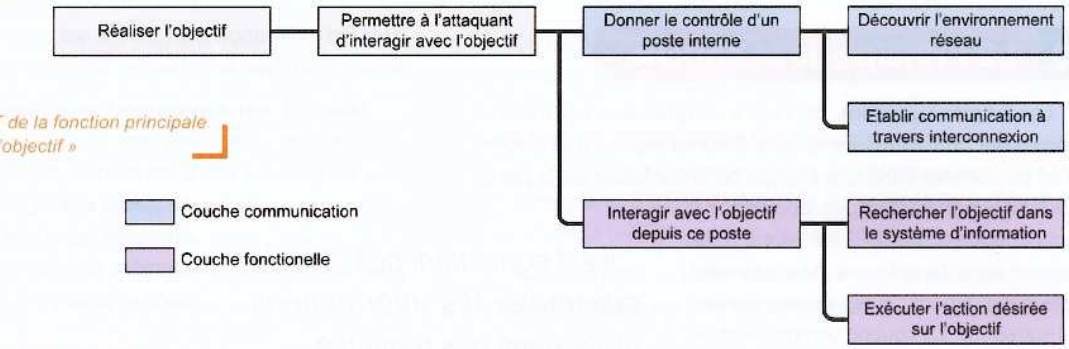
En poursuivant cette démarche d'analyse fonctionnelle, nous définissons le diagramme pieuvre représenté par la figure 2.

Au centre, nous retrouvons la backdoor interagissant avec différents éléments :

- « Attaquant » représente l'origine de l'attaque.



2 *Diagramme pieuvre de la backdoor*



3 *Diagramme FAST de la fonction principale « Réalisation de l'objectif »*

- « Objectif » représente l'objectif, le vol de données dans notre exemple.
- « Outils de protections » représente les différents logiciels de surveillance et détection qui pourraient déceler l'attaque.
- « Utilisateurs » représente les utilisateurs du système informatique, notamment ceux du ou des postes piratés.
- « Forensics » représente l'éventuelle équipe d'analystes qui tentera de déterminer la source et l'objectif de l'attaque si celle-ci est détectée.

Nous pouvons ensuite définir les différentes fonctions :

- FP1 : la fonction principale : permettre à l'attaquant de réaliser son objectif.
- FC1 : l'attaque ne doit pas être détectée par les outils de protection.
- FC2 : l'attaque ne doit pas être décelée par les utilisateurs. Cette contrainte se distingue bien de la précédente, car les méthodes de détection diffèrent. Un humain repérera par exemple facilement un dossier portant un nom anormal alors que cette notion sera difficile à implémenter dans un outil.
- FC4 : en cas de détection de l'attaque, la backdoor doit au maximum protéger l'anonymat de l'attaquant et la nature de la cible. Nous considérerons pour cela que l'équipe menant l'investigation dispose de toutes les données créées sur les disques durs par la backdoor, ainsi que de l'intégralité du trafic réseau.

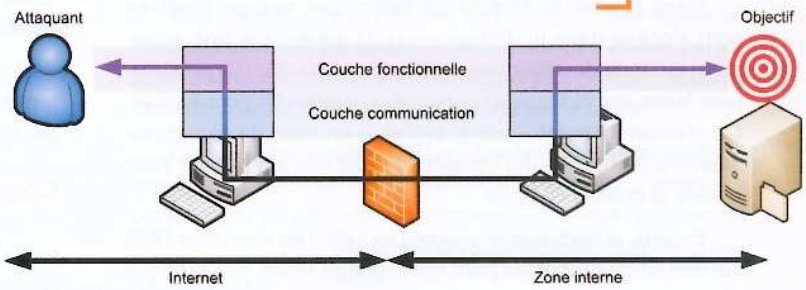
Nous en déduisons le diagramme FAST de la fonction principale, représenté dans la figure 3.

La réalisation de l'objectif se traduit par « permettre à l'attaquant d'interagir avec l'objectif », qui se découpe en deux services :

- offrir à l'attaquant un accès distant sur un poste du réseau interne ;
- permettre à l'attaquant d'interagir avec l'objectif à partir de cet accès.

Deux parties différentes apparaissent : la couche « communication » assurant la communication entre l'attaquant et le poste piraté, et la couche fonctionnelle établissant le lien entre l'attaquant et l'objectif. Cette architecture est résumée dans la figure 4.

4 *Architecture logicielle de la backdoor*



4. La couche « communication »

4.1 Rôle de la couche

En dehors de cas extrêmement simples, la backdoor ne peut atteindre l'objectif de manière autonome et doit être pilotée par l'attaquant à distance. Elle a donc besoin d'établir une communication interactive avec lui. En parallèle, l'attaquant souhaite probablement transférer des fichiers entre son ordinateur et le poste piraté.

Deux types de communications apparaissent donc :

- une communication synchrone où la backdoor récupère une commande et envoie des résultats (volume faible mais régulier) ;
- une communication asynchrone pour transférer des fichiers (volume potentiellement important).

La couche « communication » expose une API implémentant ces services.

4.2 Principe de la couche

Les communications doivent s'adapter aux contraintes définies par l'infrastructure réseau. Celle-ci impose tout d'abord de se positionner dans une logique où la backdoor est la partie « cliente » établissant des connexions vers l'attaquant, partie « serveur ». Elle impose ensuite que les flux passent systématiquement par les serveurs relais de l'entreprise. La couche communication doit en conséquent être capable :

- de découvrir un minimum la topologie du réseau afin de déterminer les protocoles autorisés, ainsi que les adresses des relais : proxy web et (éventuellement) FTP, serveur mail, serveur DNS ;
- d'exploiter ces canaux pour communiquer avec l'attaquant.

4.3 Découverte de la topologie du réseau

La découverte de l'environnement réseau repose sur une combinaison de plusieurs techniques.

Tout d'abord, la backdoor analyse la configuration des applications « par défaut » de l'utilisateur. Par exemple, si le navigateur par défaut est Internet Explorer, les coordonnées du proxy web sont obtenues en appelant des fonctions standards de `wininet.dll` ou en interrogeant directement la base de registres.

Dans le cas de Firefox ou Netscape, ces paramètres sont stockés dans un fichier `prefs.js` situé dans une sous-arborescence du répertoire `Application Data\Mozilla\Profiles` pour Netscape et `Application Data\Mozilla\Firefox` pour Firefox. Ce chemin incluant – entre autres – un nom de répertoire aléatoire (le profil), une recherche récursive est utilisée pour obtenir le chemin complet.

Ensuite, la backdoor envoie des requêtes de résolution DNS sur des noms probables pour les serveurs relais, par exemple **[proxy/smtp/mail].mycompany.com**.

Enfin, les paramètres obtenus sont validés par quelques tests (par exemple une requête HTTP vers Google).

4.4 Communication avec l'attaquant

La backdoor dispose maintenant d'une liste de protocoles autorisés et des coordonnées des relais associés. Elle peut donc ouvrir un canal de communication avec l'attaquant.

Les données peuvent être envoyées en exploitant directement les possibilités du protocole. Par exemple, pour le SMTP, l'exfiltration d'un fichier est accomplie en envoyant un simple mail avec le fichier en attachement.

Mais, il est également possible de dissimuler les informations utiles dans des requêtes et de les envoyer à un faux serveur qui saura les extraire et les reconstituer (concept de canal caché). Ce principe revient à implémenter une surcouche spécifiant la manière dont les données sont cachées dans le protocole.

...il est également possible de dissimuler les informations utiles dans des requêtes...

Communication par canal caché HTTP

Le protocole HTTP est autorisé au niveau de l'interconnexion dans la quasi-totalité des entreprises. Utilisé par de nombreuses applications, il est de plus à l'origine d'un trafic relativement important. Il est donc le premier protocole à exploiter pour communiquer avec l'attaquant.

Les communications devront transiter par le proxy. Pour garantir une bonne furtivité, il est par conséquent indispensable d'implémenter un canal caché.

Le proxy n'analyse que peu d'options de la requête HTTP pour exercer son rôle d'intermédiaire et n'effectue pas de contrôle sur les autres champs. Les données à envoyer à l'attaquant peuvent être facilement dissimulées :

- dans les paramètres transmis dans l'URL (requête GET) ;
- dans le corps de la requête (méthode POST) ;
- dans les options de la requête HTTP.

Pour ne pas dégrader la furtivité du canal caché, la quantité de données envoyées à chaque requête doit rester relativement limitée (de l'ordre de la centaine d'octets).

Le canal retour peut être aisément implémenté en cachant les données utiles dans le corps de la réponse HTTP, généralement de taille assez importante.

Communication par canal « CONNECT »

Le protocole HTTPS est également très souvent autorisé au niveau de l'interconnexion. Du point de vue de la backdoor, les possibilités ouvertes sont beaucoup plus intéressantes qu'un simple canal caché HTTP dans un tunnel SSL.

En effet, le support de ce protocole implique le proxy autorise l'usage de la méthode CONNECT. La backdoor peut alors ouvrir directement un canal chiffré sur le serveur de l'attaquant :

- ouverture des deux connexions TCP entre le client et le proxy, et le proxy et le serveur via une requête CONNECT ;
- établissement du tunnel SSL entre le client et le serveur (facultatif) ;
- envoi et réception des données utiles. Le proxy n'exerce alors plus aucun contrôle sur les données échangées, puisque celles-ci sont censées être chiffrées. Il n'est donc pas indispensable de respecter la norme HTTP, même si le tunnel SSL n'a pas été établi.

Communication par canal caché DNS

L'implémentation d'un canal caché dans DNS est également possible. Cette technique manque cependant à mon sens de furtivité.

Communication par SMTP et FTP

Le protocole SMTP permet d'établir un canal unidirectionnel très efficace pour envoyer des données de taille importante de la backdoor vers l'attaquant. Le protocole FTP apporte directement un canal bidirectionnel.

Choix du protocole

Chaque protocole a ses spécificités qui le rendent plus ou moins adapté au transport d'un bloc de données, en fonction du type de communication et de la taille de ce bloc : le canal caché HTTP sera parfait pour assurer les communications synchrones, tandis qu'un simple mail sera beaucoup plus efficace pour exfiltrer un gros fichier. La backdoor doit donc intégrer un minimum d'intelligence pour choisir le protocole le plus adapté.

4.5 Analyse des fonctions contraintes

4.5.1 Contournement des outils de détection

Les communications entre la backdoor et l'attaquant peuvent être détectées au niveau du poste piraté et au niveau de l'interconnexion.

Détection sur le poste piraté

La détection sur le poste piraté vient essentiellement des outils type firewall personnels. Dans l'absolu, ces logiciels interceptent toute tentative d'accès au réseau effectuée par un programme inconnu. Les communications initiées par la backdoor sont donc détectées.

Plusieurs points viennent cependant mitiger ce résultat. Tout d'abord, les postes internes d'une entreprise sont rarement équipés de tels logiciels. Relativement lourds, parfois instables, perturbant l'administration à distance ou le fonctionnement des applications, affichant des messages incompréhensibles, ils sont plutôt mal perçus tant des utilisateurs que des administrateurs.

Ensuite, il existe des techniques pour les contourner (comme l'exécution sous forme de *thread* injecté dans un processus autorisé). Le site « *firewall leak tester* » [2] contient une série de programmes de test exploitant une série de techniques et présente un tableau résumant le résultat pour différents produits.

La backdoor doit être capable d'exploiter certaines de ces techniques pour contourner ces éventuelles protections.

Détection au niveau de l'interconnexion

Au niveau de l'interconnexion, la détection peut être faite en comparant, à partir d'une série de critères, le trafic généré par un client par rapport à un trafic « légitime ».

Par exemple, le trafic généré par une consultation web a des caractéristiques très particulières :

- asymétrique : le rapport *upload/download* est très faible ;
- en rafale : l'accès à une page implique le téléchargement de plusieurs ressources en parallèle ;
- irrégulier : une fois la ressource obtenue, l'utilisateur la consulte et ne génère plus de requêtes.

Pour obtenir une bonne furtivité, la couche communication doit donc intégrer un contrôle des flux pour maîtriser le débit et l'allure du trafic généré.

4.5.2 Invisibilité pour les utilisateurs

Les connexions réseau ouvertes par la backdoor sur les différents serveurs sont listées par exemple par une simple commande `netstat -ano` et peuvent éveiller les soupçons d'un utilisateur un peu averti.

La backdoor peut utiliser des techniques d'exécution (présentées dans la suite) sous forme de thread injecté ou de *rootkits* pour masquer ses communications.

Deux points doivent être cependant relevés :

Tout d'abord, ces techniques se basent sur des appels de fonctions avec des paramètres spécifiques, peu utilisés par les applications « normales ». Elles augmentent donc le risque de détection par un outil d'analyse installé sur le poste.

Ensuite, il est relativement rare que les utilisateurs (lambda) surveillent en temps réel le trafic envoyé ou les connexions ouvertes par les différentes applications. Le risque de détection par une analyse manuelle reste donc très limité.

La backdoor peut utiliser des techniques d'exécution ...

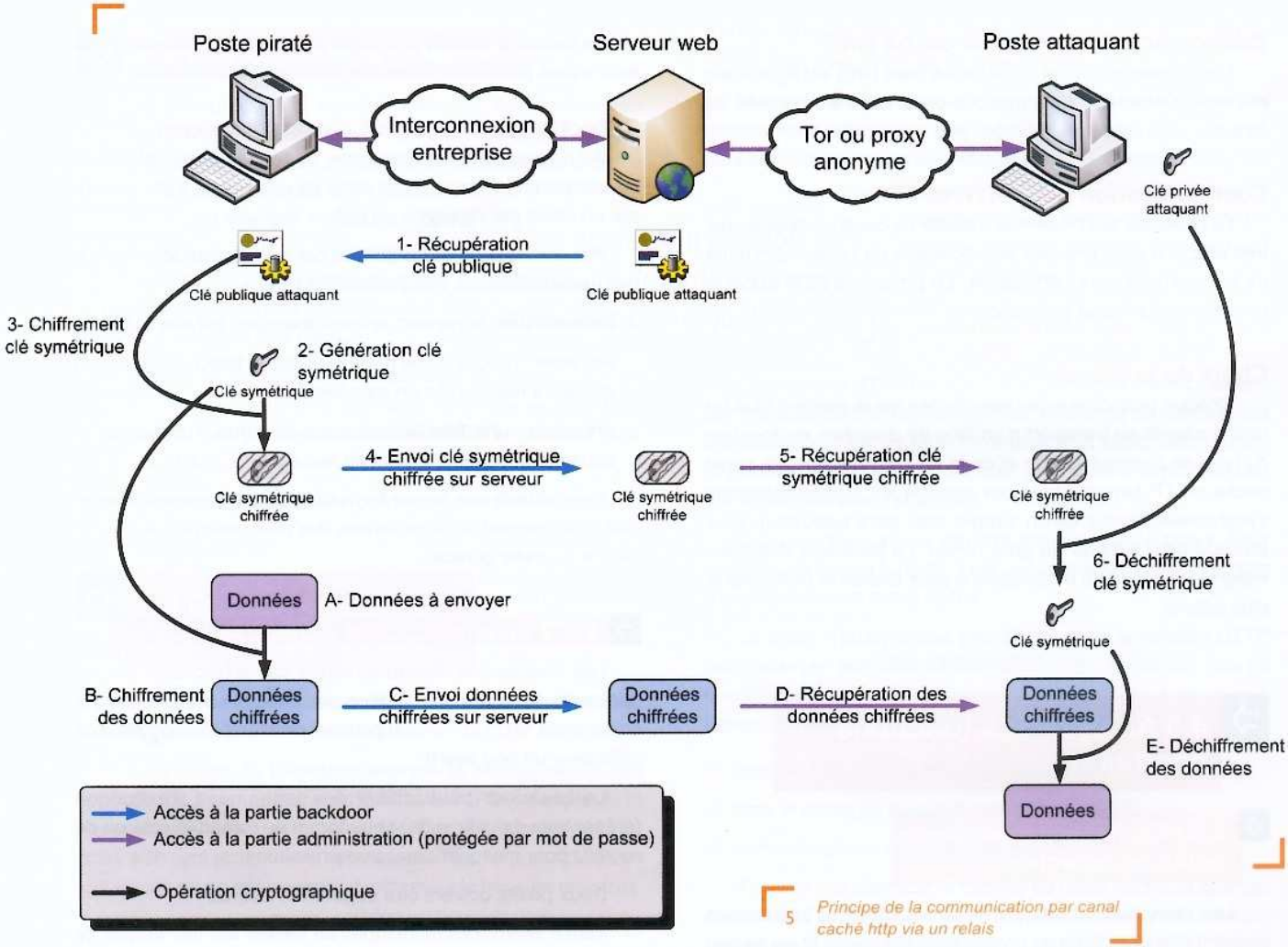
L'utilisation de ces méthodes dépendra donc du contexte technique et humain de l'attaque. Dans certains cas, il sera préférable de n'intégrer aucun mécanisme pour masquer les communications réseau aux utilisateurs.

4.5.3 Résistance à l'analyse post-mortem

Si l'attaque est découverte, deux informations primordiales doivent être préservées : l'anonymat de l'attaquant et la nature de la cible.

Préserver l'anonymat de l'attaquant

Les différentes requêtes envoyées par la backdoor contiennent toutes une adresse destination qui référence plus



5 *Principe de la communication par canal caché http via un relais*

ou moins directement l'attaquant : adresse IP, nom DNS dans les requêtes HTTP, adresse mail dans le protocole SMTP, etc.

La protection de l'anonymat de l'attaquant peut se faire en insérant systématiquement des relais entre la backdoor et l'attaquant pour tous les protocoles.

Protéger la nature de la cible

La nature de la cible peut être découverte en analysant le trafic réseau ou en récupérant des données stockées sur les relais. Pour empêcher cela, toutes les données émises doivent être systématiquement chiffrées. Comme dans toute application utilisant la cryptographie, apparaît alors le problème de gestion des clés.

Illustrons ces aspects avec deux exemples pratiques : les communications par HTTP et par SMTP.

En pratique : cas des communications par canaux cachés HTTP

Les relais sont de simples serveurs web. Le site web hébergé comporte deux parties : la première traite les requêtes provenant de la backdoor (récupération des données envoyées et stockage dans un fichier local) ; la seconde, protégée par une authentification web classique, correspond à la partie interrogée par l'attaquant (récupération des données précédemment stockées et envoi à l'attaquant).

Ce site est implémenté avec seulement quelques pages de PHP. Il suffit ensuite de le déployer chez un quelconque hébergeur, si possible dans un pays sans relation diplomatique avec celui de l'entreprise ciblée. L'attaquant récupère les données en se protégeant au travers de proxies, de sites ouverts (réseau *wifi* ou autres) ou encore en passant par le réseau Tor (ces approches n'étant pas exclusives).

Une couche de cryptographie est ajoutée pour répondre à la problématique de la protection de la nature de la cible. L'attaquant dispose à l'origine d'un couple clé publique/clé privée. Le chiffrement se base sur le mécanisme suivant :

- La première étape est de partager un secret (clé symétrique) entre la backdoor et l'attaquant. La backdoor commence par générer une clé symétrique aléatoire et la chiffre à l'aide de clé publique de l'attaquant (qui est directement embarquée dans

l'exécutable ou a été récupérée à partir d'un serveur web via le canal caché HTTP). Le résultat est envoyé et stocké sur un serveur web. Un peu plus tard, le pirate se connecte sur la partie attaquant du site, récupère la clé symétrique chiffrée et utilise sa clé privée pour la déchiffrer.

- Ensuite, toutes les données échangées entre la backdoor et l'attaquant sont chiffrées avec la clé symétrique avant d'être envoyées et stockées sur le serveur web.

La figure 5 résume ces différentes étapes dans le cas où la clé publique est récupérée par le canal caché.

En pratique : cas des communications par SMTP

L'anonymat de l'attaquant est garanti en passant par un réseau de *remailers*. La nature de la cible est préservée par un chiffrement PGP.

Les quelques lignes de *batch* suivantes donnent une idée du mécanisme qui peut être utilisé :

```
@rem Importation des clés pgp du remailer et de l'attaquant
gpg.exe --import remailer.asc
gpg.exe --import attaquant.asc

@rem Chiffrement du document " confidentiel.doc " avec la clé PGP
     de l'attaquant
gpg.exe --trust-model=always --yes -e -a -r attaquant@yahoo.fr
        confidentiel.doc

@rem Formation du mail
more header.txt > mail.txt
more notes.doc.asc >> mail.txt

@rem Chiffrement du mail avec la clé PGP du remailer
gpg.exe --trust-model=always --yes -e -a -r anon@remailer.hastio.
        org mail.txt
```

Le fichier `header.txt` contient simplement :

```
::
Anon-To: attaquant@yahoo.fr
```

Le mail final envoyé à anon@remailer.hastio.org est :

```
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.6 (MingW32)

hQIOA4zdadrEp8TtEAf9Gu8r4pBUsbKKuGy2iWJF+3Bs0yEkh4vN4qW/kwf2B1r9
…
```

Seuls l'adresse mail du remailer et le message chiffré peuvent être récupéré au niveau du serveur mail de l'entreprise. L'obtention du fichier `confidentiel.doc` nécessiterait d'avoir à la fois la clé PGP du remailer et celle du pirate.

L'attaquant pourra récupérer le mail par un accès en HTTPS au webmail (en utilisant toujours des mécanismes d'anonymisation).

4.6 Structure de la couche « communication »

En résumant l'analyse précédente, d'un point de vue opérationnel, la backdoor doit :

- échanger des données suivant différents protocoles avec un serveur sur Internet ;
- assurer une bonne furtivité de ces communications, c'est-à-dire :
  - choisir le protocole le plus adapté pour réaliser l'envoi de données ;
  - assurer un contrôle des flux pour gérer le débit et maîtriser le profil du trafic généré ;
- intégrer un mécanisme de chiffrement des données.

La couche communication est structurée suivant cinq couches décrites succinctement ci-dessous. Même si cette comparaison comporte bien évidemment des limites, il est intéressant d'introduire un parallèle entre cette pile et le modèle OSI.

4.6.1 La couche connexion

Cette couche gère l'établissement de connexions avec un serveur : ouverture d'une connexion TCP simple ou d'un tunnel « CONNECT » à travers un proxy web, établissement d'un tunnel SSL. Du point de vue de la backdoor, il s'agit de connexions « point à point », même si celles-ci transitent par un proxy.

Cette couche s'apparente donc à la couche physique du modèle OSI.

4.6.2 La couche protocole

La couche « protocole » assure le transfert de données entre la backdoor et un relais via une connexion « point à point ».

Actuellement, quatre moteurs sont disponibles :

- « Protocole HTTP » implémente l'échange de données par canaux cachés dans HTTP. Les données émises par la backdoor sont dissimulées à la fois dans les paramètres transmis (`GET` ou `POST`) et dans les options de la requête. Les données reçues sont cachées dans des pages HTML à un offset aléatoire après encodage en base 64, pour que les pages formées ne contiennent que des caractères imprimables.
- « Protocole DIRECT » implémente un protocole « maison » constitué simplement d'un en-tête et de données optionnelles.

- « Protocole SMTP » et « Protocole FTP » implémentent l'envoi de données via les protocoles SMTP et FTP.

Certains protocoles gèrent un mécanisme de retransmission en cas d'erreur.

La taille maximale des données pouvant être transmises par un unique échange à ce niveau varie en fonction du protocole. Par exemple, il est possible d'envoyer une centaine d'octets par le protocole HTTP et plusieurs milliers par le protocole DIRECT.

Cette couche s'apparente à la couche liaison du modèle OSI.

La séparation entre les couches « protocole » et « connexion » offre une souplesse intéressante permettant d'implémenter rapidement différents types de communication : une communication par canal caché HTTP via proxy transparent sera traitée par la combinaison « connexion TCP » + protocole HTTP. Si le proxy supporte la méthode CONNECT et que nous souhaitons une communication efficace, sans mécanisme de furtivité et chiffrée, nous pouvons utiliser la combinaison « connexion CONNECT » + canal SSL + protocole DIRECT.

4.6.3 La couche paquets

Cette couche assure le transport d'un bloc de données de bout en bout. Elle implémente plusieurs services exécutés séquentiellement lors de l'envoi de données.

Service « choix du protocole » (routage)

La couche sélectionne un protocole de transport en fonction de la taille des données à envoyer.

Service « contrôle du flux »

La couche analyse ensuite les spécificités du protocole choisi en termes de contrôle de flux. Par exemple, le protocole HTTP impose un contrôle strict afin que le profil du trafic généré soit proche d'un trafic HTTP légitime : les communications sont en rafale sur un même serveur pendant une durée très limitée et en établissant plusieurs connexions simultanées.

Au contraire, le protocole DIRECT n'impose aucun contrôle particulier : les données sont envoyées immédiatement.

Service « fragmentation »

Les requêtes au niveau de la couche « protocole » ne peuvent contenir qu'une quantité limitée de données utiles, qui peut être bien inférieure à la taille du bloc total à envoyer. La couche « paquets » doit donc réaliser une adaptation transparente en fragmentant les données à envoyer suivant une taille dépendante du protocole choisi.

Les fragments sont envoyés par différents relais, voire par différents protocoles. Le réassemblage est donc effectué au niveau du poste de l'attaquant.

Une demande d'envoi d'un bloc de données peut par conséquent se traduire en une série de requêtes au niveau protocole, comme représenté sur la figure 6.

Cette couche regroupe les services réseau et transport du modèle OSI.

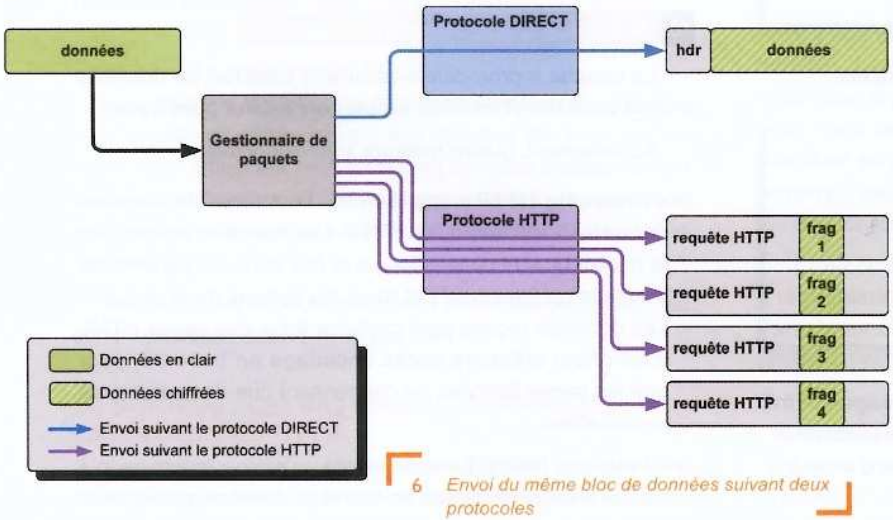
4.6.4 La couche cryptographique

Cette couche assure le chiffrement et éventuellement la compression des données à envoyer. Elle s'apparente à la couche présentation du modèle OSI.

4.6.5 La couche application

Cette couche implémente l'interface et assure la synchronisation entre les N threads représentant la couche fonctionnelle et le service de communication.

La figure 7 résume la structure de la pile réseau de la backdoor.



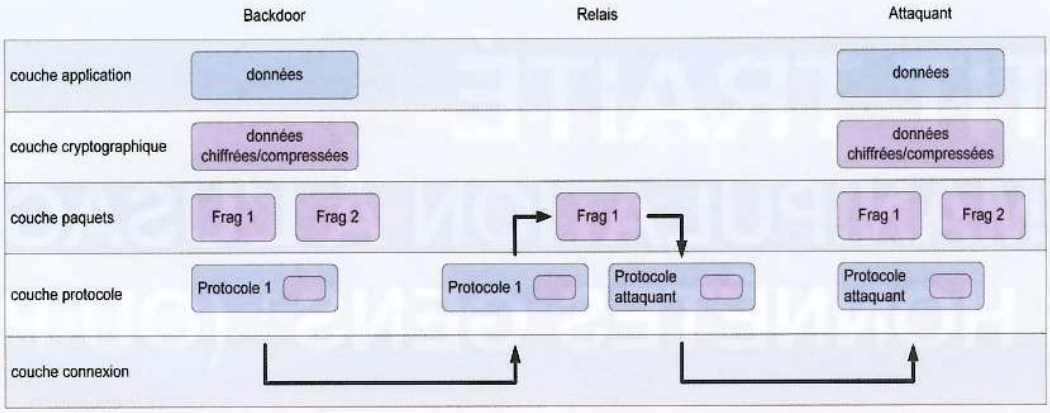
6 *Envoi du même bloc de données suivant deux protocoles*

4.7 Implémentation de la couche « communication »

4.7.1 Structure générale

La fonction d'envoi de données est utilisée par plusieurs threads en parallèle. Deux implémentations sont envisageables pour cette couche :

- Sous la forme d'une simple fonction : chaque thread gère alors son propre pool de connexions réseau. L'implémentation est par certains aspects relativement rapide, mais la synchronisation des threads pour



7 *Pile réseau de la backdoor*

l'accès aux données partagées reste lourde et il est difficile d'obtenir un contrôle de flux efficace et donc une bonne furtivité.

⇨ Sous la forme d'un véritable gestionnaire de paquets, formé d'un thread unique gérant des files de requêtes envoyées/à envoyer/en attente. La difficulté de programmation est accrue, mais il est possible d'obtenir un véritable contrôle des flux, voire une gestion de la priorité des paquets (et pourquoi pas de la QoS...)

4.7.2 Modularité

Avant l'attaque, le pirate a une certaine idée des flux autorisés par l'interconnexion et peut en fonction personnaliser la backdoor. Il est cependant possible qu'il découvre de nouvelles possibilités de communication lors de l'attaque et souhaite les exploiter. L'intégration dès la conception de tous les protocoles possibles augmente inutilement la taille de la backdoor et n'est pas une solution satisfaisante.

Il est préférable d'adopter une structure modulaire et de gérer les protocoles sous forme de modules. Concrètement, un module est une simple dll exportant une fonction par ordinal, *uploadée* depuis le serveur de l'attaquant sur le poste piraté et chargée dans l'espace mémoire de la backdoor via la fonction `LoadLibrary`. Le module est ensuite initialisé en appelant la fonction exportée. Celle-ci va enregistrer une interface protocole, constituée d'un tableau de pointeurs de fonctions standardisées dans la backdoor, ajoutant ainsi dynamiquement le support de ce protocole.

Dans une utilisation typique, seul le protocole HTTP est initialement inclus dans la backdoor. Une fois le canal de communication établi, l'attaquant peut analyser l'environnement et choisir d'utiliser le protocole DIRECT. Il uploade le module correspondant sur l'ordinateur cible, le charge dans la backdoor et l'active. La communication bascule alors automatiquement sur ce protocole.

Conclusion

Au terme de cette première partie, la backdoor développée est capable d'ouvrir un canal de communication répondant aux contraintes posées par l'analyse fonctionnelle.

Elle doit maintenant permettre à l'attaquant d'interagir efficacement avec le système d'information. L'analyse de la structure de cette couche fonctionnelle fera l'objet d'un prochain article.

Liens

[1] x90re's backdoors : page présentant le développement issu de cette étude (version bridée, vidéos), http://benjamin.caillat.free.fr/backdoors.php

[2] Firewall leak tester : site présentant le résultat d'études sur le contournement des firewalls personnels sous Windows, http://www.firewallleaktester.com/

Fred Raynal – fred@miscmag.com / François Gaspard – kad@miscmag.com

PETIT TRAITÉ D'E-MANIPULATION À L'USAGE DES HONNÊTES GENS[1] (OU PAS)

« Calomniez, calomniez, il en restera toujours quelque chose. » – Beaumarchais

mots clés : ***guerre de l'information / opération blanche / opération noire / SEO***

À l'Ouest rien de nouveau, la calomnie fonctionne encore très bien, et Internet n'a pas fait grand-chose pour améliorer la situation. Au contraire. Citons le cas [2] d'un présumé pédophile, insulté avec insistance sur un newsgroup à tel point que la police a fini par faire une descente chez lui et saisir son ordinateur. La fouille a révélé la présence d'images douteuses, et il fut incarcéré préventivement. Et pourtant, au procès, il fut acquitté.

De telles histoires ne sont – hélas – pas rares. Internet est un champ d'informations, on peut en recueillir, mais on peut aussi y semer beaucoup de choses. C'est ce que se propose d'aborder cet article au travers d'un exemple totalement fictif mais, nous l'espérons, illustratif visant une société de services. Pour cela, nous nous appuyons sur des techniques issues du référencement (SEO ou Search EngineOptimization) afin d'optimiser la présence de certaines informations auprès des moteurs de recherche.

1. Préambule

Cet article présente comment des attaques peuvent être menées sur Internet essentiellement à base d'informations. L'objectif est de montrer comment quelques personnes avec des moyens raisonnables sont susceptibles d'organiser une opération visant une entreprise, une organisation ou un État. Un précédent article paru il y a longtemps [1] décrivait une telle opération dans le monde réel à l'encontre d'un cabinet d'avocats. Cette fois, nous ciblons une SSII.

De nos jours, n'importe qui est à même de se transformer en *cyberwarrior*, et ce, principalement pour deux raisons :

- démocratisation des outils adaptés : il est simple et peu cher de créer ou de louer des armes numériques alors que réparer les dégâts s'avère dispendieux ;
- pas de droits d'entrée : pas la peine de se soumettre à l'autorité d'un autre groupe pour agir, il suffit de mener ses propres opérations.

[1] Ce titre est plagié de l'excellent ouvrage [8].

[2] Cette histoire est évidemment bien plus complexe. Nous ne la détaillerons pas plus, en outre parce qu'encore actuellement, quand on recherche dans les moteurs populaires à partir du nom de la personne et du mot-clé « pédophile », on retrouve encore de nombreux résultats, mais aucun relatif à son acquittement.

Pour monter une opération offensive, on peut assembler plusieurs « petites » actions (mêmes inoffensives), l'objectif étant par exemple de désorganiser le système dans son ensemble.

Nous avons choisi, dans l'exemple ci-après, une approche distribuée, combinant plusieurs niveaux, de préférence à une attaque centralisée. D'une certaine manière, on s'inspire également du phénomène (à la mode) de *long tail* (longue traînée) [2]. Très sommairement, le constat tactique qui en est tiré est que, parfois, il ne vaut mieux pas se concentrer sur les 5 leaders d'un secteur, mais plutôt sur tous les autres. Par exemple, en vente, une librairie conventionnelle est limitée par ses stocks, et doit donc choisir avec soin les livres qu'elle vend, d'où le choix naturel des best sellers. En revanche, un libraire sur Internet génère beaucoup plus de business en vendant tous les autres livres, que ces best sellers.

Notre objectif étant bêtement destructeur...

Dans l'univers réel, ce principe correspond assez bien au modèle de guérilla, tel que mis en œuvre en Irak par exemple. Il s'agit d'attirer tous les opposants aux « occupants américains », et de les laisser prendre des initiatives à leur encontre. Au final, les actions non coordonnées des uns et des autres (sabotage du système d'eau, explosion de pipelines, etc.) empêchent le pouvoir en place de mener ses devoirs critiques, et la population se retourne ainsi contre « l'occupant ». Sur le net, il est très facile de trouver des gens ayant une passion (ou une obsession) commune, que ce soit Star Trek ou tuer des infidèles.

On pourrait songer à appliquer ce principe de deux manières. Tout d'abord, quand la cible est clairement identifiée, afin de chercher à rassembler tous les opposants imaginables. Mais parfois, la cible est un secteur, quelque chose d'un peu flou (par exemple, une opération contre le secteur pétrolier, bancaire, etc.) : on choisirait alors d'agir non pas sur le leader du secteur, mais sur ses concurrents et de les « agréger », de les fédérer dans une action commune (ce dont il n'auront pas forcément conscience). Mais, attention, le principe fonctionne sur Internet, car la traîne est quasi infinie, ce qui n'est pas le cas dans la réalité.

Dans la suite, nous commençons par présenter les principes de la stratégie employée. Celle-ci reposant essentiellement sur de la communication sur Internet, nous présentons ensuite quelques techniques issues du monde du référencement. Nous viendrons alors à notre exemple.

2. Un peu de théorie : méthodologie d'une attaque informationnelle

Notre objectif étant bêtement destructeur, plutôt que de nous concentrer sur une seule faiblesse, nous nous appuyons sur plusieurs. On pourrait comparer cela à la méthode des pyromanes : plutôt que d'allumer un incendie à un seul endroit, il s'agit d'allumer plusieurs foyers, en tenant compte de nombreux paramètres afin de maximiser les dégâts.

L'opération se déroule en 3 étapes (après le recueil d'informations) :

1. peupler les attaquants, c'est-à-dire recruter des personnes qui agiront en fonction de l'objectif désiré, parfois sans même en avoir connaissance ;
2. préparer le champ de bataille, c'est-à-dire profiter de l'initiative pour choisir et préparer notre arsenal ;
3. exporter l'action : la plupart du temps, les attaques informationnelles reposent sur une caisse de résonance (opinion publique, places spécialisées comme les bourses, etc.).

Peupler les attaquants

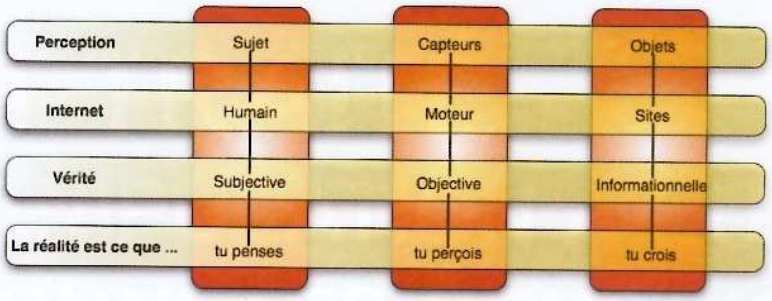
La première étape est donc de recruter des attaquants, et nous allons soit les chercher où ils sont, soit les attirer à nous :

- On rejoint les groupes de contestation, par exemple, les associations de consommateurs, des syndicats, etc.
- On crée des lieux permettant aux opposants de se rencontrer et d'échanger.

Dans le premier cas, quand on passe par Internet, on prend soin d'utiliser des techniques d'anonymisation (par exemple, Tor, des proxies, et le réseau WiFi ouvert et sans protection du fast-food du coin). Mais, on pourrait très bien participer à des groupes de contestations dans la vie de tous les jours et non plus sur Internet. Par exemple, participer à des réunions d'ONG, groupes anti-mondialisations ou encore associations de consommateurs. Cependant, il faut se déplacer physiquement à ces réunions, ce qui prend plus de temps... mais cela reste souvent fructueux.

Pour organiser et attirer la concurrence (ou les concurrents), plusieurs approches sont envisageables (et combinables bien sûr) :

- monter un site pot de miel, c'est-à-dire une source fiable pendant une longue période en se reposant sur l'impartialité et la légitimité relativement à un sujet, puis qui, une fois le public fiabilisé, fait évoluer son contenu vers l'opposition ou la rumeur (signalons que les blogs et *feeds* rss sont parfaits pour cela) ;
- monter un site d'opposition qui centralise tous les reproches faits à la cible, à ses produits, à ses comportements, etc. ;
- profitez des faiblesses de certaines ONG pour leur faire parvenir un vrai-faux rapport sur la cible qui vont alors l'utiliser contre la cible.



Le modèle de la perception appliqué au web

Peu importe les solutions retenues, dès qu'on agit sur Internet, notre objectif est d'accroître la visibilité de nos sites, d'où le recours au référencement présenté ci-après. Autre approche pour promouvoir de tels sites (à effectuer au moment opportun), le mail !

Bien souvent, il suffit de se pencher sur Google, MSN et Yahoo pour moissonner une pléthore d'adresses d'employés (voir [4]). Reste alors à prendre notre plus beau clavier pour rédiger un petit mail vantant les mérites de notre site. Accessoirement, on pourra tenter aussi les alias de mails, genre `*@target.com`, pour peu que le serveur soit mal configuré, mais l'inconvénient est alors de provoquer une surcharge anormale du serveur, qui pourrait être détectée, là où des mails envoyés normalement passeront inaperçus.

...l'objectif est toujours d'influencer la cible ...

Le champ de bataille : Internet

Dans notre exemple, nous restreignons volontairement le champ de bataille à Internet, mais les conséquences de nos actions, les réponses à nos actions, etc. pourront être en dehors. De plus, des combinaisons sur des terrains variés s'avèrent souvent intéressantes. Par exemple, lancer un procès pour violation de brevet impose au défendeur de sortir les éléments de ses recherches attestant qu'il n'utilise pas le brevet en question... mais, dans le même temps, il révèle les solutions sur lesquelles il s'appuie.

N'oublions pas non plus que nos vraies cibles sont humaines, bien plus que les ordinateurs et les réseaux qui ne sont que des moyens de parvenir à nos buts. Il s'agit donc de prendre en considération ces différentes cibles, qu'elles soient intermédiaires ou finales.

Si on reprend un modèle de perception classique, on a un sujet, muni de capteur(s) qui perçoit un objet au travers de ses capteurs. Ce modèle s'adapte également à Internet : un humain cherche de l'information (en général à l'aide de moteurs de recherche) et visite les sites alors présentés. En reprenant le modèle de perception, l'humain/utilisateur est le sujet, le site web retourné l'objet et les capteurs sont les différents moteurs de recherche.

Quelle que soit la forme de l'attaque, **l'objectif est toujours d'influencer la cible**, l'amener à entreprendre une action (ou à ne pas agir, ce qui revient au même), que ce soit conscient ou non de sa part. Ce qui différencie ces 3 attaques sont – selon les auteurs de cet article – les cibles, et en conséquences, les techniques employées. Pour nous, l'intoxication vise les capacités de raisonnement de l'adversaire, la déception sa perception et la désinformation son environnement. Une fois la technique choisie (déception, intoxication, désinformation), cela permet de savoir quels outils seront nécessaires.

Détaillons maintenant les actions d'influence à base d'information les plus classiques, en les adaptant à notre terrain de jeu :

- **Intoxication** : imaginons un site web contrôlé par l'attaquant, qui publie des articles. Une fois que la cible est assidûment accrochée (il est facile de s'en rendre compte grâce aux traces laissées par les navigateurs et autres proxies), on modifie légèrement le contenu du site, par exemple pour faire en sorte que la cible attache plus d'importance à certains facteurs, qui rentrent alors dans sa prise de décision.

- **Déception** : elle vise les moteurs de recherche dans la mesure où ils sont la paire de lunettes au travers de laquelle la majorité des gens voient Internet... sauf que ces lunettes sont déformantes.

- **Désinformation** : connue depuis des années sur le net avec tous les canulars (hoax), les rumeurs se propageant depuis les forums et chaînes de mails, etc.

[3] Le page rank est une sorte d'indice de confiance d'une page, développé par Google. L'objectif des entreprises de référencement est souvent d'avoir le meilleur page rank possible, car synonyme de visibilité sur Internet.

Signalons une particularité de notre environnement : les moteurs de recherche sont à la fois des capteurs, mais aussi des éléments de l'environnement. Ils sont une cible à la fois de déception, mais aussi de désinformation. Il s'agit de modifier le comportement normal du moteur. À une époque pas si lointaine, il était par exemple possible de voler le sacro-saint *page rank*[3] de sites, en gros en mettant une redirection bien comme il faut du site d'origine vers le site pirate. Nous revenons par la suite sur ce genre de techniques.

Cette partie de l'attaque est la plus longue, car elle suppose de préparer pas mal de matériel : articles, rapports, sites web, etc. souvent, avant même que l'attaque ne débute. Cela peut (voire doit) être conduit en même temps, quand ce n'est pas en avance, que le recrutement des acteurs. En outre, c'est dans cette phase que les informations collectées au préalable serviront à alimenter les attaquants qui disposent également de leurs propres munitions que nous ne contrôlons pas, mais qui doivent, dans la mesure du possible, être intégrées dans notre stratégie générale.

Exporter l'action

Une fois la tactique décidée, l'objectif est généralement de rendre publique l'action, pour lui donner plus de poids, d'ampleur (et continuer à recruter des participants dans la mesure du possible).

Nous cherchons alors à promouvoir notre propre contestation et à diminuer l'écho des réponses de la cible. Comment ? Citons Google :

> *Q : What can I do if I'm afraid my competitor is harming my ranking in Google?*
>
> *A : There's almost nothing a competitor can do to harm your ranking or have your site removed from our index.*

Et si ce n'était pas aussi fiable... comme nous le verrons.

3. Référencement sur Internet

Le référencement sur Internet, plus communément appelé en anglais « *Search Engine Optimization* » (SEO), est une technique bien connue des développeurs de sites web. Ce n'est pas tout de créer un site web, il faut également que celui-ci soit le plus visible possible. C'est ici que les techniques de SEO interviennent. Le but ultime est d'être le mieux classé dans les résultats d'un moteur de recherche lorsqu'un Internaute effectue une requête sur un ou plusieurs mots-clés.

Il est bien connu que, lors d'une recherche, les internautes ont tendance à consulter un site web classé dans le top 5 plutôt qu'un site web classé à la 20ème position. L'idée est donc d'obtenir le plus de visibilité sur Internet. Avant d'entrer dans des techniques plus sombres, nous commençons par un bref rappel des techniques de base pour bien référencer un site web sur Internet.

Le but ultime est d'être le mieux classé dans les résultats d'un moteur de recherche...

- les mots-clés : être le plus créatif possible, éviter les mots-clés généraux (ceux que tout le monde tend à utiliser) et les mots-clés poisons (viagra, casino...), voire utiliser aussi des mots-clés mal orthographiés ;
- avoir une bonne architecture pour les liens entrants/sortants/internes (éviter les liens sortants vers des sites pornographiques par exemple) ;
- mettre à jour le contenu assez souvent : plus le site change de contenu plus les moteurs de recherches ont tendance à les référencer (pensez aux sites de news) ;
- avoir du contenu innovant, pas simplement recopié sur les autres sites web ;
- choisir intelligemment son nom de domaine et le titre de ses pages ;
- etc.

Cet article n'étant pas un cours sur le référencement web, nous nous arrêterons donc là. L'important est d'avoir une idée générale des techniques des webmasters pour référencer au mieux un site web. Il est évident qu'il n'est pas toujours facile de suivre tous les bons conseils et que même en les suivant cela prend parfois des mois pour qu'un site commence à être bien référencé.

Mais viennent alors les techniques moins orthodoxes pour référencer son site web, celles que les moteurs de recherches n'aiment pas particulièrement, celles appelées « *Black Hat Search Engine Optimization* ».

Le Black Hat SEO est en général défini comme l'utilisation des techniques que les moteurs de recherches n'aiment pas dans le but d'être bien classé dans les résultats.

Il est à noter qu'il n'y a ici rien d'illégal. Le terme Black Hat peut porter à confusion puisqu'il est aussi utilisé dans le milieu de monde de la sécurité informatique pour nommer un groupe de personnes (plutôt méchantes). En SEO, il en est tout autre. Comme précisé, il s'agit simplement de techniques que les moteurs de recherches n'aiment pas.

Les deux principales raisons d'utiliser le Black Hat SEO sont d'une part d'améliorer la visibilité de son site et d'autre part de profiter des systèmes de PPC (*Pay Per Click*). Le PPC sortant largement du cadre de notre sujet et n'étant d'aucune utilité ici,

nous ne nous attarderons pas dessus. Voyons plutôt quelques techniques de base en Black Hat SEO.

Black Hat SEO : cloaking

Le *cloaking* est certainement la technique la plus connue. Le but est de changer le contenu d'une page web en fonction des paramètres des visiteurs. Un des buts est d'être bien indexé avec un contenu adapté, mais quand un internaute arrive sur la page celle-ci affichera quelque chose d'autre. Dans le cadre qui nous intéresse (les attaques informationnelles), on pourra imaginer référencer une page avec des informations légitimes sur la cible, mais, quand l'internaute arrive sur celle-ci, il découvrira tout un tas d'informations complètement au désavantage de celle-ci. Par exemple, on peut créer un site référence avec des informations légitimes concernant une société, mais quand un internaute arrive sur le site, les informations seront complètement différentes (fausses connexions avec des réseaux occultes, délinquances fiscales, etc.).

Il existe plusieurs types de cloaking. S'ils ont tous le même but, tous ne fonctionnent pas de la même façon. Mais surtout, certains deviennent trop facilement détectables par les moteurs de recherches (eh bien oui, les moteurs de recherches essayent de détecter le cloaking).

User-Agent cloaking

Le plus ancien et simple cloaking est le *User-Agent Cloaking*. Lors d'une requête HTTP, un des champs les plus intéressant est le `User-Agent`. Dans le cas d'un robot web, et plus particulièrement Google, ce champ a une valeur similaire à `"GoogleBot"`. Il est facile de savoir que la requête provient d'un robot web et non d'un Internaute. Le petit script PHP suivant renvoie alors le robot vers une page de notre choix :

```
$flag=strpos($_SERVER["HTTP_USER_AGENT"],"Googlebot");
if ($flag) {
  include("googlebot-special.html");
} else {
  // afficher page normale
}
```

Cette technique n'est pas très difficile à utiliser, cependant elle est presque devenue inutilisable. En effet, il est particulièrement facile de fausser la valeur du champ `User-Agent`. Un robot web pourrait venir un jour avec un certain `User-Agent` et le lendemain avec un autre. Notre petit script PHP ne serait alors plus d'aucune utilité. Il ne faut pas non plus oublier qu'une fois qu'un moteur de recherche détecte que vous « trichez », il y a de grandes chances que votre site ne soit plus référencé du tout. C'est vraiment la dernière chose souhaitée.

Referer cloaking

Une autre technique similaire à la première est d'utiliser le champ `referer`, qui indique d'où provient un internaute (en d'autres mots, s'il est arrivé sur notre site web en cliquant sur un lien d'un autre site, s'il a effectué une requête dans Google, etc.). On peut donc ici filtrer sur les mots-clés cherchés par l'utilisateur :

```
if (isset($_SERVER["HTTP_REFERER"])) {
  $referant = strtolower($_SERVER["HTTP_REFERER"]);
  if ((strpos($referant, "http://www.google.")!==false)
  && (strpos($referant, "q=israel")!==false)) {
     header("Location: http://www.pro-hezbollah.com");
     exit();
  }
}
```

IP cloaking

La dernière manière présentée pour effectuer du cloaking repose sur l'adresse IP. Encore une fois, cette adresse pourra être retirée à partir d'un champ de la requête HTTP (`REMOTE_ADDR`) :

```
$ip  = strval($_SERVER["REMOTE_ADDR"])
```

Cette méthode est la plus efficace des trois, car il est beaucoup plus difficile de masquer une adresse IP. Cependant, elle est la plus compliquée à implémenter, puisqu'il faut tenir à jour une liste de toutes les adresses IP des *web crawlers*.

Autres techniques du Black Hat SEO

Toujours améliorer son pagerank...

Une autre technique efficace pour augmenter le nombre de liens entrants (ou *backlinks*) est simplement d'ajouter des commentaires intelligents sur des *guest books*, blogs ou autres forums. Le commentaire contiendra en plus un lien vers votre site web. Si le contenu du commentaire est sans aucun sens, il y a beaucoup de chance qu'il soit effacé, un système de post automatique n'est donc pas la meilleure des idées.

...ou comment diminuer celui de ses concurrents ?

Dans la catégorie « je veux ennuyer mon concurrent », une des astuces est d'utiliser le *keyword poisoning*. L'idée est simplement d'injecter des *keywords poisons* sur le site de son concurrent. Ces mots sont supposés être mal vus par les moteurs de recherche qui pénalisent les sites les employant pour

se référencer. Il est bien évident que le site du conçurent doit *permettre de poster des entrées de la part d'un utilisateur : forum*, blog, guest book ou autre.

Autre technique, le *Google Bowling*. Cette technique, qui est une des plus connues, est de créer le plus possible de mauvais liens vers le site de votre cible. Tous les sites à caractères sexuels, jeux en lignes, sites racistes, Viagra, etc. sont bon à utiliser. Accroître le nombre de mauvais liens vers un site diminue la qualité de son référencement.

Encore plus direct, on peut utiliser le *Google Washing*. Ici, on ne parle plus de liens, mais de dupliquer tout le site de sa cible. Seul le nom de domaine sera (légèrement) différent. Les moteurs de recherche n'aiment pas les contenus dupliqués et auront tendance à bannir un site. À vous maintenant de faire en sorte que le site banni ne soit pas le vôtre, mais celui de votre cible. En général, seulement un des sites est banni et c'est souvent le site le plus récent qui l'est. Une bonne idée est donc de racheter un très vieux nom de domaine et de l'utiliser comme Google Washing... ou d'anticiper.

> *Le « spamouflage » est encore une autre technique permettant d'infliger des dégâts au site de la cible.*

Pour ceux qui ont beaucoup de temps (et on y reviendra par la suite), il est possible de créer un site tout à fait légitime avec du contenu de qualité sur un certain sujet. Une fois le site bien référencé (et premier dans la liste) et ayant un certain crédit, on *change le contenu : technique dénommée Google Insulation.*

Le « spamouflage » (spam + camouflage) est encore une autre technique permettant d'infliger des dégâts au site de la cible. On poste un message sur un blog ou autre et on inclut tout un tas de mauvais liens, mais, dans la liste de mauvais liens, on y inclut aussi le site cible. Il n'est pas toujours évident de savoir comment les moteurs de recherche vont réagir dans ce cas, mais il est arrivé qu'ils bannissent toutes les adresses d'un coup. Ça vaut toujours le coup d'être tenté.

Après cette brève introduction, nous nous contenterons de citer quelques techniques comme Black Hole SEO, 302 Page Hijack, Blogger Bowling, Black Hat Blog and Ping... Pour le lecteur intéressé, deux sites sont à ne pas manquer pour se tenir au courant des (presque) dernières nouveautés en matières de SEO : bluehatSEO (**http://www.bluehatseo.com/**) et seoblackhat (**http://seoblackhat.com/**). À noter que ce dernier contient un forum payant, mais que le blog est gratuit.

La plupart de ces approches n'ont rien d'illégales. Mais, quand on commence à combiner cela avec des « exploits » plus habituels au sens de la sécurité informatique : utiliser des *cross site scripting* pour rediriger le site de la cible vers des casinos online ou des sites pornos. Mais ceci est une autre histoire...

4. L'attaque en elle-même

Il va de soi que tout ce qui est présenté dans la suite est totalement fictif, qu'aucune des situations décrites n'est inspirée par la réalité...

Dans un premier temps, nous présentons les acteurs, la situation, le contexte. Ensuite, nous donnons les grandes lignes de la stratégie de l'attaquant. Les deux dernières parties portent respectivement sur des opérations blanches et noires. Pour mémoire, ces opérations viennent en complément de la trame principale, pour renforcer certains effets. Dans ces deux cas, nous insisterons sur les aspects techniques, bien souvent négligés (l'informatique n'est vu que comme conteneur). Pourtant, nous verrons comment, entre autres grâce au SEO, la technique peut renforcer l'opération.

Nous supposerons que l'attaquant a déjà recueilli les informations nécessaires à la planification de son opération, ceci afin de nous concentrer sur la tactique mise en place pour l'attaque elle-même.

Les acteurs

Commençons par présenter les acteurs. L'opération est initiée par une entreprise de services informatiques (indienne par exemple) qui souhaite racheter une entreprise du même genre, mais basée en Europe, en France en particulier. Pourquoi ? Surtout pour acquérir son carnet d'adresses. Nous appellerons cette entreprise « Proctor » pour plus de facilité. Nous nous plaçons dans la peau de l'entreprise indienne. Le but de l'opération est donc de racheter cette entreprise pour avoir accès à son réseau de relations.

La stratégie

Le principe général est de distendre les liens qui existent entre Proctor et son carnet d'adresses, c'est-à-dire ses clients. Toutefois, nous ne nous en prendrons pas directement à eux. L'axe principal de l'opération vise à saturer le service commercial de Proctor.

Notre stratégie a une double motivation. D'une part, on souhaite racheter plus facilement (ou à plus bas prix). D'autre part, on aimerait que Proctor dépense son énergie dans cette lutte pour le maintien de son réseau, car c'est mauvais pour ses affaires. Ceci lui coûte de l'énergie (peu importe si c'est en termes humain, financier ou autre) et c'est bien ça qui importe : cette énergie dépensée pour récupérer ces liens ne sera pas dépensée pour autre chose (comme contrer le rachat).

Double jeu : se méfier de la mariée

Cette partie de l'opération en constitue le cœur. La société indienne contacte Proctor en lui proposant de collaborer sur un marché étranger à son terrain. Proctor est une société internationale, mais essentiellement basée en France, pas mal en Europe et aux États-Unis. Bien consciente que l'Asie représente un énorme marché en développement, et qu'une partie de son activité sera de plus en plus externalisée entre autres en Inde, il s'agit pour notre société indienne de proposer une première collaboration enrichissante. Pour Proctor, c'est un gain financier et la perspective de pénétrer un nouveau marché. Mais Proctor n'est pas dupe et sait très bien que cette proposition n'est pas uniquement philanthropique. En contrepartie, la société indienne réclame... la même chose : être co-traitant sur des marchés européens.

L'attaquant ne révélera ses véritables intentions que lorsqu'il sera trop tard...

À ce moment-là, tout semble merveilleux. L'attaquant ne révélera ses véritables intentions que lorsqu'il sera trop tard, au moment d'achever (acheter) Proctor.

Du point de vue de l'attaquant, quels bénéfices ? Tout d'abord, cela permet d'étudier Proctor de l'intérieur, de trouver les personnes-clés, les processus, etc. Ensuite, en proposant à Proctor de venir sur le marché indien (ou Sud-Est asiatique), Proctor doit concentrer des ressources (commerciales et juridiques en particulier) qui seront bien accaparées. Par exemple, pendant les négociations du partenariat, il est très probable que Proctor sollicite ses propres juristes, mais aussi un cabinet externe. Demander régulièrement des amendements (mineurs bien sûr), changer des propositions, etc., ce qui prend du temps à chaque fois.

Une fois cette collaboration scellée juridiquement, Proctor doit répondre à un gros marché conjointement avec la société indienne (et inversement) : la force commerciale est en route. Dans un premier temps, pour que le mariage ait lieu, il faut proposer une vraie bonne affaire. Mais ensuite... de fausses bonnes affaires conviendront tout à fait. Idéalement, une affaire qui consomme beaucoup de ressources, mais avec une marge très faible (l'intérêt pour Proctor étant qu'elle lui permet de vraiment s'implanter plus sur le marché indien), voire ensuite des marchés qui se compliqueront avec des querelles juridiques. Tous ces marchés se doivent d'être gros, conséquents pour accaparer un maximum de personnes chez Proctor (qui de son côté doit – c'est le contrat qui a été scellé – faire la même chose pour la société indienne en Europe, sauf que les 2 n'ont pas la même capacité d'embaucher, voir ci-après).

...nous allons faire voir la vie en rose au service commercial.

Focus : droguer les commerciaux ou déception du marié

Dans le même temps, nous allons faire voir la vie en rose au service commercial. Sommairement, comment fonctionne un service commercial ? Un commercial dispose d'un carnet d'adresses, passe des coups de fil, essaye d'obtenir un rendez-vous. Quand il y parvient, il discute avec son potentiel client pour lui faire accoucher de quelques informations, comme les besoins à venir, histoire de les anticiper (voire de rédiger pour lui le cahier des charges). Dans le même temps, il est supposé disposer d'une veille sur les appels d'offre qui sortent à droite à gauche.

Chez Proctor, ça se passe presque comme ça, sauf que les commerciaux sont essentiellement des juniors : ils sont soumis à une forte pression, et leur veille des appels d'offre est proche de 0. Autre spécificité relevée pendant notre phase de recherche d'informations, les différentes entités de Proctor sont très cloisonnées et échangent peu les unes avec les autres.

Quelle tactique adopter ? Rendre les commerciaux H-E-U-R-E-U-X !!! Pour cela, c'est très simple, il nous suffit de leur fournir tout ce dont ils ont besoin :

- **des contacts** : il s'agit de récupérer une liste de clients potentiels, puis de la faire parvenir aux commerciaux de Proctor, sauf que cette liste n'est pas directement exploitable (préciser les noms, trouver les adresses email ou les numéros de téléphone, etc.).

  Pour récupérer la liste, on peut s'appuyer sur un cabinet de relations publiques (qui servira également plus tard). Il n'aura pas été choisi au hasard, car il participe aussi régulièrement à des salons professionnels. En conséquence, il dispose de la liste des visiteurs. Idem si notre société indienne investit dans quelques salons du même ordre : elle finira par obtenir une telle liste (et en plus aura gagné en visibilité, ce qui sera intéressant pour le développement après le rachat).

  Reste maintenant à transmettre la liste aux commerciaux. Dans une preuve de bonne foi, la société indienne peut la céder à Proctor en indiquant qu'il s'agit de secteurs à haut potentiel. Autre approche, on commence par cibler quelques commerciaux de Proctor, puis successivement, on les « invite » à participer à des réunions pour qu'ils présentent l'entreprise. On spécifie bien qu'ils n'ont pas le droit de venir avec un ordinateur, mais uniquement une clé USB (ou équivalent). Ils doivent alors la brancher sur un portable où traîne malencontreusement un fichier `liste clients.xls`. Sans vouloir mettre en cause la moralité de tous les commerciaux, il est fort probable que quelques-uns se laissent tenter par ce fruit défendu.

- **des appels d'offre** : profitant de la pression et du manque de temps et de moyens des commerciaux junior, nous réalisons la veille sur les appels d'offre qu'ils n'ont pas le temps de faire. Quand on en trouve, on leur envoie : un simple mail de la part de Monsieur Durant, service achat, de la société réellement éditrice de l'appel d'offre peut suffire. Mais, Monsieur Durant tient vraiment à ce que Proctor réponde à l'appel d'offre. Pour s'en assurer, il le transmet à plusieurs personnes. Pour chaque appel d'offre, Monsieur Durant dispose de 3 choix :

1. Il ne le fournit qu'à un commercial, dans le service réellement susceptible de répondre.
2. Souvent, plusieurs services sont susceptibles de répondre dans une même entité : il le communique à tous, les laissant régler les comptes entre eux, les arbitrages étant faits en interne de Proctor laissant des traces sur les commerciaux qui finalement ne porteront pas la réponse (puisqu'ils n'auront alors pas la commission qui va avec).
3. Proctor est divisé en plusieurs entités, réparties géographiquement sur plusieurs régions. Lors de l'analyse initiale, l'attaquant s'est rendu compte qu'en dessous de certains montants il n'y avait pas de recoupement d'information. En envoyant l'appel d'offre à ces différentes entités, on provoque alors des réponses de plusieurs parties de Proctor, qui ne se seront pas concertées, ce qui est toujours du meilleur effet.

des salaires plus importants... ailleurs : imaginons qu'on passe par un cabinet de recrutement pour tenter de débaucher quelques commerciaux-clés, en faisant miroiter des salaires autrement plus élevés, des primes, etc., le tout étalé sur quelques nombreux entretiens, ça donnera des velléités de départ à certains ou, tout au moins, ça contribuera aussi à miner le moral ambiant, déjà que la pression est élevée avec tous les appels d'offre et le développement à l'international !

La mise en place d'un tel site n'est ni triviale, ni immédiate...

Pour résumer, il s'agit de saturer le service commercial, soit de l'interne via un nouveau partenariat, soit de l'externe par des offres et autres qui donnent l'impression à ce service de fonctionner au mieux alors qu'il est sous perfusion (perfusion qu'il sera toujours temps d'arrêter un peu plus tard, juste avant de négocier le rachat... et s'il y a moins d'affaires qui rentrent, ça baissera encore le prix).

Cette partie de l'opération repose sur la saturation d'un service en lui fournissant trop d'informations, celles qu'il ne parvient pas à trouver tout seul, et comme il n'a pas la capacité de les traiter... ce qui demande des capacités de recherche performantes, chose qui n'est pas si simple qu'il paraît. D'une certaine manière, les commerciaux sont les capteurs de Proctor, et on les aveugle (ils risquent de rater les appels auxquels ils répondaient habituellement pour se concentrer sur les nouveaux, plus avantageux) : on est donc dans le domaine de la déception.

Opérations blanches complémentaires

Dans cette partie, nous nous concentrons tout d'abord sur les attaques à base informatique. L'intérêt des systèmes d'information vient de leur dualité, à la fois en tant que conteneur et contenu. Généralement, les actions sont ciblées sur l'un ou l'autre des aspects. Nous utilisons, dans ce qui suit, les deux conjointement.

Rappelons aussi que toutes les opérations décrites ci-après sont menées dans un même timing, bien difficile à représenter ici, bien que crucial. Certaines actions sur un terrain ont vocation à renforcer celles menées sur un autre (cf. la partie sur la stratégie générale).

Intoxication via la promotion d'un site web

Comment procéder pour arriver à ce résultat ? Commençons par monter un site de contestation. La mise en place d'un tel site n'est ni triviale, ni immédiate.

Lors de la phase recherche d'informations [4], nous avons collecté énormément d'informations concernant Proctor, mais également sur l'ensemble du secteur d'activité. Plutôt que de monter un site à charge contre Proctor, nous élaborons un site de référence sur le travail dans les sociétés de services, par exemple avec notation/évaluation, une sorte d'évaluateur pour tous les acteurs du marché (ça tombe bien, un tel site n'existe pas, c'est une très bonne opportunité pour nous).

De tels évaluateurs existent dans le monde financier, par exemple le SRI (*socially responsible investing*) : ils prennent en compte des facteurs éthiques, financiers, humains, structurels, etc., pour attribuer une note à des entreprises. Charge à nous de créer un indicateur pour le secteur qui nous intéresse, indicateur qu'on contrôle et qui pourra donc (dé)favoriser les éléments de notre choix. De cette manière, notre site apparaîtra neutre dans un premier temps puisqu'on référence tous les acteurs. Lors de la phase de démarrage, on veille à ne pas dévaloriser Proctor (restons neutre). On profite ici du fait qu'un tel site n'existe pas (il y a juste des forums où des anciens employés expliquent leur vision de la vie dans les SSII, ce qui nous servira par ailleurs).

La création et l'installation d'un tel site prend un certain temps (ou un temps certain). Il va surtout falloir faire connaître notre site, et le légitimer par une utilisation communautaire (l'indépendance pour être crédible). Les techniques classiques de SEO, voire Black Hat SEO, se révèlent utiles. En outre, on envoie un mail au personnel des sociétés évaluées pour les informer de la création du site (les adresses email sont faciles à collecter sur Internet [4]).

Toujours pour améliorer notre visibilité, on contacte des sites de nouvelles dans le domaine, comme ZDNet ou 01. Dans un premier temps, l'envoi d'un communiqué de presse permet d'informer les journalistes, puis nous (enfin, notre représentant, par exemple un cabinet de relations publiques agissant pour nous sans connaître la finalité de l'opération) proposons des interviews.

Pour être bien référencé sur Internet, un des secrets est aussi de proposer un vrai contenu : fabriquons-le. On fonde l'évaluation des SSII sur plusieurs critères. Par exemple, l'aspect ressources humaines, très souvent décrié dans les SSII, qu'on enrichit avec des articles issus d'employés (anciens ou non). Comment les trouver ? Via les annuaires des écoles d'ingénieurs par exemple ou des sites de réseaux sociaux.

Via le cabinet de relations publiques, on cherche également à prendre contact avec les clients des SSII : ils sont affichés sur le site de la SSII. En général, il s'agit de clients satisfaits. Mais, en cherchant dans les archives des sites (par exemple sur archives.org), on peut déterminer certaines des références qui apparaissent et surtout disparaissent.

Les six premiers mois, nous restons le plus neutre possible concernant le secteur. Nous voulons juste attirer le plus grand nombre de personnes sur notre site et obtenir une certaine crédibilité. On veillera attentivement aux logs de notre site web pour savoir d'où proviennent les utilisateurs, leur fréquence de visite, ainsi que les pages qui les intéressent.

Pour augmenter les chances que le site soit bien vu de la part des internautes (qui peuvent être des gens curieux, des clients de Proctor ou autres) et possède une certaine neutralité, on crée un forum favorisant un échange de discussions, ou un blog, pour être plus *trendy* web2.0. Quel que soit le médium retenu, il sera important de le modérer en toute rigueur, pour accroître encore notre image d'impartialité. Par exemple, « on » aura posté un message très agressif envers Proctor. Assez « rapidement » (comprendre : on laisse quelques heures, le temps que le message soit lu, et que quelques personnes commencent à réagir), le modérateur (c'est-à-dire nous) intervient de deux manières :

1. On modère le message, c'est-à-dire qu'on le supprime du médium.

2. On poste un message citant en partie ce message agressif en expliquant que ces allégations gratuites ne sont pas tolérées.

Ainsi, on fait d'une pierre deux coups. D'une part, la calomnie est semée, et souvenons-nous de la citation de Beaumarchais reprise en début d'article. D'autre part, notre impartialité apparaît renforcée, et donc la confiance générale envers le site également.

Après quelques mois, quand notre site dispose d'un certain crédit auprès de ses visiteurs et d'une bonne visibilité sur Internet, il est temps de diffuser des articles moins en faveur de Proctor. Attention cependant à ne pas être complètement négatif, mais à incliner progressivement notre ligne éditoriale (sans d'ailleurs se focaliser trop sur Proctor pour que ça ne soit pas perceptible).

Nous disposons maintenant d'un bon outil d'influence, notre site vers lequel les personnes intéressées sont dirigées grâce à son bon référencement. Mais outre le fait de disposer d'un vecteur d'influence, quel est l'autre intérêt de ce site ? Identifier les acteurs qui pourront nous aider dans notre opération. En effet, nous avons les moyens d'identifier les internautes hostiles à Proctor.

Enfin, pour conclure, revenons sur la méthodologie initiale en 3 étapes :

- **peupler les attaquants** : le « recrutement » de contributeurs sur le site augmente d'autant son importance tout en nous donnant les éléments d'informations nécessaires à une attaque informationnelle ;
- **préparer le champ de bataille** : grâce à diverses techniques de SEO, nous mettons notre site en avant ;
- **exporter l'action** : après avoir fourni le terrain et les informations sur notre site, les relations publiques et autres journalistes déplacent et amplifient notre message.

L'idéal est que Proctor ait connaissance de ce site une fois qu'il est bien installé pour qu'il le surveille, voire qu'il cherche à y répondre (que ce soit sur le site même ou par d'autres canaux), toujours dans la perspective d'épuiser les ressources de la cible.

Proctor sur le web : nettoyage par le vide

Le site web d'intoxication est loin d'être suffisant. Restons dans notre champ de bataille, Internet, mais essayons maintenant de tirer profit des moteurs de recherche. Nous exploitons certaines faiblesses des moteurs pour attaquer le site web de la cible à l'aide de Black Hat SEO. Pour impliquer le plus de ressources possibles de Proctor, cette attaque démarre quand notre site d'intoxication commence à changer de ligne éditoriale.

Notre objectif est de diminuer le page rank du site web de Proctor. Il se trouve que Proctor vend un service, service qui est aussi vendu par d'autres sociétés (étrangères ou non). Lorsqu'un internaute cherche des informations sur ce service, le premier lien retourné par les moteurs de recherche est Proctor. Nous allons faire en sorte que cela change, d'une part en améliorant le classement des concurrents de Proctor, et d'autre part en diminuant celui de Proctor. Nous combinons plusieurs approches (pourquoi se priver) :

Le site web d'intoxication est loin d'être suffisant....

- **Google Bowling** : nous cherchons à créer un grand nombre de backlinks vers Proctor. On utilise des forums, des blogs, des guest books... à caractère pornographique, raciste ou type casino en ligne et autres ventes de Viagra.

  Plus efficace encore, on crée de tels sites dans lesquels on insère « malencontreusement » des liens et des mots-clés voisins ou identiques à ceux de Proctor. Il faut bien réaliser que construire automatiquement des sites pornographiques est extrêmement facile : peu de texte et beaucoup d'images,

largement disponibles sur des sites web et des newsgroups. Il suffit d'écrire un petit programme qui les assemble, et les présente sous une thématique commune.

Enfin, ou pourra également s'appuyer sur des sites *blacklistés* (sites qu'on réalisera nous-mêmes pour s'auto-blacklister ou qu'on identifiera à partir du croisement des résultats de multiples moteurs de recherche).

> *...nous n'avons rien utilisé d'illégal jusque-là, toutes les techniques sont dites « blanches ».*

- Google Washing : on duplique le site de notre SSII, on rachète un très vieux nom de domaine (si on n'en a pas en stock, il n'a pas besoin d'être en rapport avec notre cible) et on copie à l'identique le site de la SSII. Il est bien sûr possible d'effectuer cette opération de multiples fois afin de diluer d'autant le site de la cible.

- Création d'une *link farm* avec du contenu en rapport avec la cible, bon ou mauvais, et, sur chacun des sites, on place des liens vers le site web de Proctor.

Ces différentes actions génèrent de l'activité autour du site web de Proctor, tout en portant préjudice à sa bonne image. À noter que nous n'avons rien utilisé d'illégal jusque-là, toutes les techniques sont dites « blanches ».

En conclusion, on attaque l'image de Proctor sous deux angles : en augmentant la visibilité de notre site de contestation et en diminuant celle du site web de la cible à l'aide du SEO.

Mais, encore une fois, ces deux actions sont loin d'être suffisantes, et viennent en complément à une stratégie générale pour la renforcer : comme l'objectif est le rachat de Proctor, s'en prendre à son image permet de négocier le prix au mieux.

Des techniques moins orthodoxes

Jusqu'ici, nous avons fait attention à ce qui était employé. Une question arrive cependant : et si nous couplions ce type d'attaque avec des attaques dites « de pirates informatiques » ? Les deux attaques précédentes visent essentiellement l'image de l'entreprise, mais combinons cela avec des actions à l'encontre de son fonctionnement interne.

« hacking » for profit

Pourrir un système d'informations est vraiment très simple. À titre d'exemple, citons un petit florilège de possibilités : utilisation de botnets, compromission de serveurs DNS, modification de la configuration réseau (gardent-ils les configurations des routeurs Cisco), contrôle des emails, effacement des serveurs sensibles (genre un contrôleur de domaine, sachant que le serveur de sauvegardes est malencontreusement tombé en panne la veille), une fuite d'information, un serveur DHCP concurrent qui apparaît sur le réseau, etc. Les possibilités sont nombreuses et variées, mais il faut encore disposer d'une personne de confiance pour mener une telle opération, ce qui n'est peut-être pas le cas de notre attaquant indien. Ceci dit, avec le mercenariat...

Plaçons-nous maintenant dans un tel contexte. Avant tout, une remarque préliminaire sur l'attaque à la mode, le déni de service. Il n'apporterait rien à notre cause. Outre le bruit, ça risque d'éveiller l'attention de Proctor : qui voudrait faire un DDoS (en dehors d'un adolescent pré-pubère ou d'un mafieux russe) ? À la place, l'attaquant préfère une action discrète, furtive.

Tout d'abord, il est primordial de connaître parfaitement la topologie du réseau afin de mieux le cibler. Ensuite, il s'agit d'en prendre le contrôle. Passons sur le moyen par lequel notre pirate réussit à obtenir un accès (infiltration, faux recrutement, etc.) dans la mesure où ce n'est pas très difficile pour une cible telle que Proctor (gros turnover favorisant les entrées sorties, pas de gardiennage). Imaginons qu'il ait accès à un ordinateur portable issu de l'entreprise. Il commence par l'analyser pour y trouver :

- le mot de passe de l'utilisateur ;
- le mot de passe d'admin local ;

La fouille de cet ordinateur pourrait révéler de nombreuses autres informations [5, 6]. Quoi qu'il en soit, en 2-3 jours (voire heures – si, si, c'est possible...), il dresse un panorama du réseau, des serveurs de fichiers, d'impression, aux back up, mais surtout les contrôleurs de domaine (et en particulier les comptes d'admin de domaine). L'expérience du test d'intrusion montre qu'il n'est souvent pas besoin d'exploits ou autres prouesses techniques pour prendre le contrôle complet du réseau, juste un peu d'imagination (histoire de deviner les mots de passe) et de temps (histoire d'être discret et de casser les hash des mots de passe).

Bref, une fois la main sur le réseau, toutes les machines n'ont pas le même intérêt. Commençons par le serveur de messagerie. Comme nous souhaitons connaître exactement ce qui se passe en interne, il constitue un nœud crucial. Déjà, il contient une foule d'informations intéressantes. En plus, il nous permet d'identifier des personnes importantes, que nous pourrons contacter pour en faire des *insiders* ou à surveiller à cause de leur rôle stratégique. Il est possible également de nuire bien plus :

- On pourra organiser une fuite d'information depuis le compte d'un employé précis, ce qui nuira autant à l'employé qu'à l'entreprise (par exemple, la base client d'un commercial ou les documents confidentiels transmis à un chef de projet par un client important).

- Annuler l'émission et la réception de certains emails (si possible importants, mais l'intrus ne va pas passer son temps à surveiller

tous les mails, donc le faire aléatoirement sera aussi bien) : les mails n'étant pas transmis, les relations avec l'extérieur... mais aussi en interne vont souffrir.

Le contrôle des emails offre une multitude d'options sans pour autant être quelque chose de très visible (attention toutefois à la manière dont l'information acquise est utilisée pour ne pas révéler l'existence de la fuite).

Une autre possibilité est de contrôler le serveur DNS ou le proxy web. On analyse alors les fréquentations de sites par employé, ce qui peut aider dans le cadre d'un profil (voir article sur ce thème dans ce dossier). En plus, on peut en profiter pour rediriger aléatoirement de temps en temps quelques requêtes vers notre site d'intoxication.

Le contrôle des emails offre une multitude d'options...

Enfin, profitons d'avoir la main sur le réseau pour adapter la visibilité de Proctor. Nous profitons ici du fait que cette entreprise héberge elle-même son site web pour adapter un peu de SEO : mettons en place notre propre cloaking. L'idée est, selon l'origine d'une requête, de diriger vers telle ou telle page. En installant un module qui va bien (LKM, *backdoor*, etc.), soit sur le DNS, soit sur le serveur web lui-même, on redirige à volonté le trafic vers le site de Proctor. En fonction de la source de la requête (que ce soit DNS ou web), on renvoie vers qui/où on veut. Par exemple, on pourrait proposer le vrai site pour les connexions venant de l'Intranet, mais un site modifié vu de l'extérieur. Ceci dit, cette approche est risquée, car de nombreux employés sont en prestation à l'extérieur : il est probable que l'un finisse par remarquer les anomalies. En revanche, on peut cibler plus finement qui on redirige : les meilleurs clients (avec le risque évoqué précédemment), les gens qui ont comme *referer* un site de recrutement (voir le focus ci-après), etc.

Tant qu'à faire également, puisque les moteurs de recherche n'aiment pas le cloaking, on prendra soin de le faire maladroitement pour eux afin de se faire détecter, et de perdre le page rank.

Les SSII ne sont pas réputées pour le côté « humain »...

Concrètement, cela n'est pas aussi compliqué qu'il y paraît. Un simple module noyau jouant sur les `skbuff` sous Linux ou agissant sur la couche NDIS des Windows conduit à ce résultat. Quelques centaines de lignes de code sont suffisantes pour installer la redirection. Étonnamment, la mise en place des pages alternatives, avec subtilité, est sans aucun doute ce qui demandera le plus de réflexion et de travail.

Focus sur l'attaque des Ressources Humaines : l'humain, le maillon faible ?

Les SSII ne sont pas réputées pour le côté « humain », au contraire de ce que mettent généralement en avant leur site web (qui se ressemblent tous avec les mêmes mots-clés).

En revanche, dès qu'on cherche 5 minutes, on tombe sur des sites comme **munci.org** ou des forums comme sur **hardware.fr** où la vie de l'intérieur est décrite. Le moins qu'on puisse dire, c'est que ça ne donne généralement pas envie.

On le sait, toutes les SSII sont pareilles, et le recrutement fonctionne de manière industrielle pour compenser un turnover élevé : quand une telle entreprise annonce qu'elle va recruter 4000 personnes alors qu'elle a seulement 15000 employés, on peut se demander quelles sont les prévisions de départs... Nous nous en prenons donc à ce processus critique pour notre cible. Nous allons jouer sur deux axes pour semer la confusion : encourager les départs, complexifier les embauches.

Commençons donc par le débauchage. Une fois des employés importants repérés (consultants, commerciaux, etc.), on est à même de leur fournir des opportunités, par exemple en les redirigeant vers des pages réelles proposant un travail similaire, mais avec plus d'avantages. Cependant, ceci n'est pas suffisant : même s'ils voient ces offres, ils sont peut-être timides et n'oseront pas contacter les personnes. Nous allons le faire pour eux. En profitant de l'accès au réseau, nous récupérons dans le SI des Ressources Humaines les fiches des employés, qui fuira malencontreusement ensuite dans la nature, par exemple vers d'autres SSII concurrentes ou sur Internet. Une telle fuite présente deux avantages pour notre opération. D'une part, elle renforce le doute à l'égard du sérieux de Proctor. D'autre part, si des employés sont débauchés, ça baissera d'autant le prix d'achat.

Du côté du recrutement, les sites sur lesquels les personnes des RH vont chasser sont bien connus (par ex. : Monster). Varions les plaisirs. Il est assez simple, via le DNS, de temps en temps de rediriger le trafic vers un de ces sites de recrutement vers une machine qu'on contrôle annonçant qu'une opération de maintenance est en cours. Toujours pour épuiser les recruteurs, on peut plus simplement créer des profils bidons directement sur le site (ça s'automatise très bien avec un bon script et des mots-clés bien choisis) : les recruteurs partiront ainsi à la chasse aux fantômes. Plus complexe, puisque l'attaquant contrôle le système d'informations, on accède aux CV récupérés par les recruteurs (stockés sur les ordinateurs personnels ou sur un partage) pour modifier tout ce qui ressemble à un numéro de téléphone et à une adresse email pour accentuer encore la chasse aux fantômes.

En complément à ces approches, la fourniture de quelques éléments d'information sert également aux deux objectifs de l'opération (débauchage et recrutement), éléments récupérés pendant la phase de recherche d'informations. On l'a déjà mentionné, qu'un *executive director* soit en pleine recherche d'emploi n'est pas super motivant pour les troupes (il faut dire que la direction vient de nommer un co-executive director qui lui rogne la moitié de ses responsabilités). Par ailleurs, sur l'Intranet

des RH (accessible de tout Internet, ça va de soi puisque les employés sont disséminés partout), une enquête d'opinion interne révèle que tous les employés se trouvent très largement sous-payés. C'est le grief principal fait à l'encontre de Proctor. Soufflons sur les braises. Une rapide recherche nous donne ce qu'il faut : le *Figaro Économie* a publié une étude sur les salaires par secteur, et la répartition entre parts fixes et variables. Effectivement, Proctor est loin de ce qui se fait, à la défaveur de ses employés. On trouve un document similaire à l'APEC. On peut alors s'appuyer sur notre site de contestation pour faire un dossier (à charge, mais sans le dire) contre la politique salariale chez Proctor : d'une part, ça ne donnera pas envie d'y entrer, et, d'autre part, ça donnera envie à ceux qui y sont soit de se faire augmenter, soit de partir voir ailleurs.

Pour terminer sur ces techniques moins orthodoxes, mentionnons que les variations sont infinies, mais bien plus risquées. Elles s'avèrent particulièrement intéressantes, tactiquement, pour récupérer de l'information et désorganiser en interne la cible, Proctor. En outre, il faut anticiper les réactions et se prémunir d'un éventuel retour de bâton. En conséquence, il est important de rester le plus discret possible afin de ne pas être identifié.

Conclusion

Les attaques informationnelles se produisent tous les jours, à plus ou moins grande échelle. Nous avons voulu montrer – et espérons avoir réussi à le faire – comment elles pouvaient se dérouler sur Internet, alors que c'est loin d'être le seul support. L'intérêt d'Internet est de favoriser la propagation rapide de l'information, et sa pérennité (quasi impossible d'effacer une information d'Internet). En outre, nous avons insisté aussi sur leur renforcement grâce à des spécificités techniques.

L'exemple, en partie scénarisé, montre la complexité de telles opérations. En effet, chaque élément interagit avec les autres, et il s'agit de bien mesurer ces interactions pour qu'elles renforcent l'effet désiré plutôt que de l'annuler.

Les attaques informationnelles reposent sur l'information, qu'il s'agisse d'en créer, modifier, cacher ou révéler, qu'elle soit vraie ou fausse. Certes, l'information en elle-même est cruciale, mais la manière dont elle atteint sa cible va influencer la perception qu'elle en a également : le vecteur et la forme jouent aussi un rôle essentiel. C'est bien cet aspect que nous avons également présenté, avec l'utilisation d'Internet et l'emploi de SEO.

Proctor est trop occupé à développé ses affaires pour se rendre compte de ce qui lui arrive. Une SSII est une coquille vide (dans le sens où elle ne dispose pas de connaissances propres ou de produits), et concevoir une attaque à base d'information n'est pas chose aisée. Nous avons donc opté pour une stratégie visant quelques éléments cruciaux à son bon fonctionnement : service commercial, image et ressources humaines. En mettant suffisamment de sable dans les rouages, Proctor deviendra indubitablement indienne.

Bibliographie

[1] ED-V, TD-K, « La guerre de l'information », *MISC* 3.

[2] ANDERSON (Chris), *The Long Tail*, WIRED, october 2004, **http://www.wired.com/wired/archive/12.10/tail.html**

[3] BRASSIER (Marc), *Le lobbying sur internet*, *MISC* 17.

[4] RAYNAL (Fred), GASPARD (François), « Information, le nouveau nerf de la guerre ? », *MISC* HS 1.

[5] PILON (Arnaud), « Sécurité des secrets du poste nomade », *MISC* HS 1.

[6] AUMAITRE (D.), BEDRUNE (J. B.), CAILLAT (B.), « Forensics, quelles traces se dissimulent malgré vous sur votre ordinateur ? », **http://esec.fr.sogeti.com/FR/documents/seminaire/forensics.pdf**

[7] IWOCHEWITSCH (Michel), « Les failles humaines et l'information... », *MISC* 34.

[8] JOULE (Robert-Vincent), BEAUVOIS (Jean-Léon), *Petit traité de manipulation à l'usage des honnêtes gens*, ISBN : 2-7061-1044-9.

Victor Stinner

COMMENT RÉALISER UN FUZZER ?

Le fuzzing est une technique d'assurance qualité logicielle visant à rechercher des bugs. Pour y arriver, on injecte des données ni tout à fait valides, ni tout à fait invalides : ce sont les cas limites que personne n'aime traiter et qu'on préfère ignorer pour simplifier le code. Dans la mesure du possible, nous nous concentrerons sur les bugs pouvant aboutir à une vulnérabilité.

mots clés : *fuzzing / format de fichier / protocole / génération de données / couverture de code*

Petite histoire du fuzzing

Au début, le *fuzzing* consistait à injecter des données totalement aléatoires (**[MILLER]**) et à constater les plantages. La tendance actuelle est plutôt aux attaques très ciblées qui imitent le comportement normal de l'application, le tout couplé à un débogueur pour détecter les bugs subtils, tels que les accès illégaux à la mémoire qui ne déclenchent pas toujours de plantage.

Trouver les vecteurs d'entrée

Pour contrôler un programme, la première étape est d'en trouver ses points d'entrée. Il n'est pas nécessaire d'en connaître la liste exhaustive dès le début. On peut tester les différents vecteurs séquentiellement ou bien, plus intéressant, les exploiter en parallèle.

Exemples de points d'entrée : socket réseau en écoute, variable d'environnement, ligne de commande, événements souris, répertoire de travail, signaux, quota systèmes (ex : `setrlimit`), etc. On peut commencer en lisant la documentation du logiciel (lorsqu'elle est disponible). Une courte utilisation du logiciel peut donner d'autres idées.

Pour un logiciel propriétaire ou mal documenté, on peut considérer la cible comme une boîte noire. Des outils d'audit et débogueurs permettent de tracer le programme pour découvrir automatiquement les vecteurs d'entrée. Sous Linux, on peut utiliser les programmes `strace`, `ltrace`, `netstat`, `lsof`, etc.

> *Pour contrôler un programme, la première étape est d'en trouver ses points d'entrée.*

Il faut toujours se demander ce qu'on audite. Lorsqu'on audite un serveur web accessible depuis Internet via une *socket* TCP : est-il pertinent de rechercher des bugs dans le *parseur* du fichier configuration ? Un bug ne pourra être considéré comme une vulnérabilité que si l'attaquant peut modifier ce fichier et forcer le rechargement de la configuration. Il faut prendre en compte le contexte. Les vecteurs d'attaque diffèrent selon l'endroit depuis lequel l'attaquant opère et ses permissions.

La sécurité est souvent un aspect mis de côté lors du développement d'une application, par manque de temps, d'argent ou de compétence. Le fuzzing doit donc être efficace car « le temps, c'est de l'argent ».

Exemple : points d'entrée du programme file

Pour que l'exemple soit concis, nous utiliserons une cible simple : le programme file qui sert à identifier le type d'un fichier. Pour l'audit, `ltrace` a été choisi, car il peut tracer simultanément les appels aux bibliothèques et aux appels système. Exécution (abrégée) de `file` :

```
$ ltrace -S -o audit -- file image.jpg
image.jpg: JPEG image data, JFIF standard 1.02

$ egrep '(SYS_open|getenv)' /tmp/ltrace|cut -f1 -d'='
SYS_open("/etc/ld.so.cache", 0, 00)
SYS_open("/usr/lib/libmagic.so.1", 0, 00)
SYS_open("/lib/tls/i686/cmov/libc.so.6", 0, 00)
(...)
getenv("MAGIC")
getenv("HOME")
getenv("POSIXLY_CORRECT")
SYS_open("/etc/magic.mgc", 32768, 00)
SYS_open("/etc/magic", 32768, 0666)
SYS_open("/usr/share/file/magic.mgc", 32768, 00)
SYS_open("image.jpg", 32768, 00)
```

On découvre trois variables d'environnement : `MAGIC`, `HOME` ainsi que `POSIXLY_CORRECT`. Les fichiers accédés sont variés : cache du chargeur `ld.so`, bibliothèques `libmagic` et `libc`, les traductions `gettext`, fichiers `/etc/magic` et `/usr/share/file/magic.mgc`, notre fichier `image.jpg`, etc.

Trouver une vulnérabilité dans les bibliothèques `ld.so`, `libc.so` ou `gettext` (qui fait partie de `libc.so`) est très intéressant, car l'ensemble des applications Linux les utilisent. D'ailleurs, `gettext` est assez sensible au fuzzing. Bien que les traductions soient normalement recherchées dans `/usr/share/locale`, la variable d'environnement `LANGUAGE` accepte le motif `../` (sauf pour les programmes setuid). Notez que ltrace n'est pas capable de tracer l'accès aux variables d'environnement de `ld.so` ou de la `libc` (ex : `LD_PRELOAD`, `MALLOC_CHECK_`, `LANGUAGE`, etc.). Il faut parfois aller jusque dans le code source pour trouver plus d'informations.

Pour revenir au programme `file` : les dossiers `/etc` et `/usr` ne sont pas accessibles en écriture pour un utilisateur non privilégié. Les vecteurs d'entrées sont donc les trois variables d'environnement précédemment citées et le fichier source (`image.jpg`). Pour gagner du temps, on peut se référer au manuel de `file` qui explique à quoi sert la variable `MAGIC` : elle indique le chemin d'un fichier `magic` personnalisé. Testons :

```
$ MAGIC=xxx ltrace -S -o ltrace -- file image.jpg
file: could not find any magic files!

$ grep 'SYS_open.*xxx' ltrace|cut -f1 -d=
SYS_open("xxx.mgc", 32768, 00)
SYS_open("xxx", 32768, 0666)
```

En modifiant la variable d'environnement `MAGIC`, on active donc deux nouveaux vecteurs d'entrée (fichiers `$MAGIC.mgc` et `$MAGIC`).

Générer des données

Le but du jeu d'un *fuzzer* est de pénétrer le plus loin possible dans la cible avant que les données ne soient rejetées. C'est ce qu'on appelle la couverture du code : plus il y a de code exécuté, plus il y a de bugs potentiels. Le pire cas étant lorsque la cible rejette les données lors du premier test de validation.

Les trois manières les plus courantes de générer des données sont, par ordre de complexité :

- ⇨ données aléatoires ;
- ⇨ mutation : injection d'erreurs dans des données valides ;
- ⇨ génération selon un modèle, souvent proche des spécifications.

Données aléatoires

Le premier type est l'attaque typique d'une boîte noire : on la stimule pour en observer son comportement. Parfois, bien que le système étudié soit entièrement connu, on préfère le considérer comme une boîte noire pour des raisons de simplification. C'est le choix du programme **[sysfuzz]** qui réalise des appels système aléatoires. Étant donné que la recherche exhaustive est trop lente, les arguments générés par `sysfuzz` ne sont que partiellement aléatoires. Le fuzzer tente plus fréquemment des valeurs particulières comme une adresse mémoire valide en espace utilisateur, une adresse non mappée ou encore certaines valeurs remarquables comme l'entier 16 bits `0xFFFF`.

Mutations

La technique des mutations consiste à partir de données valides type, de préférence un court extrait représentatif du format, puis à y injecter des erreurs aléatoires. Le type des mutations (inversion de bit, remplacement d'un octet aléatoire, incrémentation d'un octet, insertion/suppression d'octets, etc.) et leur nombre sont délicats à configurer. Un nombre trop faible d'erreurs n'aura aucun effet, alors qu'avec trop d'erreurs, les données seront rejetées en bloc par la cible. On verra plus loin comment s'aider d'un système de notation pour paramétrer automatiquement le générateur.

Classiquement, les mutations sont réalisées sur des fichiers qui sont préparés avant d'exécuter la cible. On peut gagner du temps en modifiant la cible en cours d'exécution pour y injecter directement les erreurs. Ceci simplifie beaucoup le fuzzing sur des formats/protocoles compressés, chiffrées et/ou comportant des sommes de contrôle (CRC32, MD5 ou autre). C'est cette solution qui a été choisie **[Bunny]** : il faut recompiler son programme avec un compilateur gcc modifié pour insérer des fonctions de fuzzing.

Le fuzzer **[zzuf]** a choisi un juste milieu : il surcharge les fonctions de la **libc** (`write`, `read`, `send`, `recv`,...) pour injecter des erreurs sans modifier les données sur le disque. Cette méthode originale permet de travailler avec de gros fichiers (plusieurs gigaoctets). En cas de succès, `zzuf` affiche la graine du générateur pseudo-aléatoire, utilisable pour rejouer la session.

La quantité de mutations possibles est très importante.

La quantité de mutations possibles, même sur de petits fichiers (4 Ko), est très importante. Certains *frameworks* de fuzzing optimisent la recherche en apportant une connaissance du format de fichier. Les attaques sont alors ciblées sur les champs sensibles, en particulier les champs contenant la longueur d'un autre champ.

Modèles

Un modèle est une description d'un format de fichier ou d'un protocole réseau permettant de générer des données valides selon les spécifications. On le trouve sous la forme d'un algorithme écrit soit dans un langage de programmation, soit dans un langage dédié. Ce sont surtout les champs variables du modèle qui nous intéressent : où injecter des valeurs particulières (chaîne vide, nombre négatif, chaîne UTF-8 invalide, longueur nulle ou très grande, etc.) sans pour autant rendre les données invalides ?

L'utilisation d'un modèle est, par exemple, implémentée dans **[SPIKE]** qui utilise une grammaire basée sur des blocs : bloc fixe, chaîne variable, champ longueur, bloc de fin, etc. L'écriture de la grammaire est un travail long et coûteux, mais permet des attaques plus ciblées qui ont plus de chances de réussir.

Libre arbitre

Le problème de l'apprentissage d'un format de fichier ou protocole à un fuzzer est que ce travail est très long et sa complexité proche de celle de la cible. Le fuzzer peut alors être lui-même victime d'oublis et de bugs. De plus, le code sera fortement spécifique et non réutilisable. Certains frameworks comme **[Peach]** utilisent une représentation plus haut niveau (format XML dans le cas de Peach) pour faciliter cette tâche et encourager la réutilisation.

Notation

Plus le générateur de données est fidèle à la cible, plus sa complexité risque d'augmenter. Le risque est de reproduire des suppositions faites par les auteurs de la cible. Exemple volontairement simple pour illustrer le problème : lorsqu'on demande à un utilisateur de saisir son âge, on s'attend à ce qu'il saisisse un nombre entre 10 et 90. Que se passe-t-il si l'on saisit 0, 100, 250, -1 ou même « xyz » ? Il faut que le fuzzer conserve son libre arbitre (ex : mutations aléatoires) !

Déterminer le succès d'une session (une exécution de la cible) est une tâche délicate, car les conditions permettant de déterminer un succès (crash, déni de service, détournement du flot d'exécution, etc.) ne sont pas forcément connues, ni clairement définies. Une solution est d'utiliser une collection de sondes ayant une note pondérée. Un succès est déclaré lorsque la somme des notes dépasse un certain seuil. Quelques exemples de sondes :

- raison de la terminaison d'un programme (code de retour ou signal ayant tué le processus) ;
- dépassement d'un temps limite ;
- présence de certains motifs texte dans la sortie standard ou dans un fichier de log (« *segmentation fault* », « assertion », « *null pointer* »,...),
- charge système et consommation CPU du processus ;
- etc.

Ce système de notation implémenté dans le projet **[Fusil]** (voir plus loin) s'est révélé souple et efficace sur des cibles très variées. Néanmoins, chaque nouvelle cible nécessite un paramétrage fin des différentes sondes et leur pondération respective. Dans un cas, un dépassement du délai sera un succès (déni de service), alors que, dans un autre cas, ce type d'erreur sera ignoré car peu intéressant. Ce paramétrage pourra toujours être affiné en fonction des résultats obtenus.

Couverture de code

Pour mesurer jusqu'à quelle profondeur les données sont allées dans la cible, on peut utiliser des outils mesurant la « couverture du code », tels que `gprof` qui compte le nombre d'appels et le temps passé dans chaque fonction. On peut s'en servir comme sonde : la note augmente avec la quantité de code exécutée et lorsque certaines fonctions intéressantes sont appelées (ex : `memcpy`, `gets`, `strcpy`, etc.).

L'outil d'audit **[Valgrind]** se prête très bien au fuzzing étant donné qu'il intègre des outils de contrôle des accès mémoire et des *mutex*, mais également des outils mesurant la couverture de code. Le projet **[Flayer]** exploite justement Valgrind pour identifier précisément les erreurs.

Le danger d'une sonde comme Valgrind est qu'elle réveille le principe d'incertitude d'Heisenberg : la cible est ralentie entre cinq et vingt fois, et se comportera donc différemment d'une exécution normale. Les applications réseau sont particulièrement affectées par ce problème, mais d'autres cas peuvent se présenter.

Guider le fuzzer avec la notation

Le paramétrage d'un fuzzer peut consommer beaucoup de temps, temps que l'on aurait préféré utiliser en temps de calcul pour le fuzzing. Une solution est l'apprentissage : réutiliser les sessions précédentes pour affiner les sessions suivantes au lieu de repartir de zéro à chaque fois. Le but est que le fuzzer se dirige tout seul vers son objectif en utilisant le système de notation mis en place avec les sondes.

> *Le but est que le fuzzer se dirige tout seul vers son objectif...*

Pour le cas du générateur de données par mutation, on se servira de la note pour faire évoluer le type et le nombre de mutations. Fusil génère un facteur d'« agressivité » qui augmente progressivement de 0% (état initial, n'injecte aucune erreur) à 100% (données aléatoires). Pour la mutation, ce facteur est utilisé de cette manière :

- agressivité de 0 à 25%, opérations : mutation d'un bit ;
- 25 à 50%, ajoute les opérations : valeurs spéciales et remplacement d'un octet ;
- 50 à 100%, ajoute les opérations : insertion et suppression d'octets.

Les opérations sont de plus en plus destructrices. Bien sûr, le nombre d'opérations est également proportionnel à l'agressivité.

Le facteur d'agressivité est applicable à d'autres composants : nombre de vecteurs d'entrée exploités en parallèle, taille maximale des données, etc.

Fusil

Le framework **[Fusil]** est une implémentation sous licence libre écrite en Python des idées de cet article. L'architecture est basée sur un système multi-agent qui offre une communication par événements intuitive et limite la dépendance entre les objets pour offrir une meilleure réutilisabilité de chaque agent.

Fusil propose divers agents « actions » : création de processus, génération de données par mutation (avec apprentissage), clients et serveurs réseau (dont un serveur HTTP) ; ainsi que divers agents « sondes » : surveille syslog, la sortie standard, l'utilisation du CPU, le temps d'exécution, etc. Le module `c_tools` facilite l'écriture de programmes en langage C et leur compilation.

Les projets donnés en exemple sont de divers types : ClamAV, Firefox, gettext, Gstreamer, fonction `printf()`, appels système Linux, vim, etc. Depuis sa création, Fusil a réussi à crasher un grand nombre d'applications, telles que rpm, la libc GNU (fonction `printf`), poppler (PDF), Gimp, etc. Certaines vulnérabilités ont été gratifiées d'un identifiant CVE :

- CVE-2007-2754 : dépassement d'entier dans freetype2 (police de caractères TTF) ;
- CVE-2007-2650 : déni de services dans ClamAV (document OLE2) ;
- CVE-2007-2645 : dépassement d'entier dans `libexif` (image JPEG).

Par rapport à un fuzzer écrit rapidement pour une cible précise, Fusil a de multiples avantages. L'ensemble des opérations sont loguées sous forme textuelle. On peut s'en servir pour rejouer manuellement une session (voir l'exemple Mplayer plus bas). Chaque session a son répertoire dédié où Fusil enregistre les fichiers qu'il génère. Les succès sont conservés, tandis que les autres sessions sont automatiquement supprimées. La sortie standard du programme cible est écrite dans le fichier `stdout` et chaque session est loguée dans son propre fichier `session.log`. Pour chaque nouvelle fonctionnalité ajoutée à Fusil, l'ensemble des projets de fuzzing en bénéficient.

Exemple de plantage avec Mplayer

Nous prendrons Mplayer comme cible, car il est très sensible au fuzzing (crashe rapidement), et un fichier au format Matroska, car c'est un format rare et son parseur Mplayer est donc peu testé :

```
$ fusil -p projects/mplayer.py ~/testcase/flashmob.mkv  --max-success=1
(...)
[session 4][watch:stdout] Match pattern 'MPlayer interrupted by signal'
 (score 100.0%) in 'MPlayer interrupted by signal 11 in module: decode_audio'
[session 4][watch:process:mplayer] Process exited with error code: 1
```

En lisant les logs, on apprendra qu'il aura suffit de 320 mutations de bits pour planter Mplayer. Rejouons le plantage dans `gdb` :

```
$ gdb mplayer
(gdb) run run-0004/session-0004/flashmob-0001.mkv
Program received signal SIGSEGV, Segmentation fault.
0xb6f50bf0 in vorbis_book_decodev_add () from /usr/lib/libvorbis.so.0
(gdb) disassemble $eip $eip+1
0xb6f50bf0 <vorbis_book_decodev_add+96>:        mov    (%eax,%edx,4),%edi
(gdb) print $eax
$2 = 0
(gdb) print $edx
$4 = 0
```

Honte sur moi ! Ce n'est pas Mplayer qui a planté, mais la fonction `vorbis_book_decodev_add()` de la bibliothèque `libvorbis`. L'instruction précise qui a déclenché l'erreur de segmentation est une lecture à l'adresse mémoire zéro (`NULL`). On va recompiler la bibliothèque `libvorbis` en désactivant les optimisations gcc et en activant les symboles de débogage (`CFLAGS="-O0 -g"`). Rejouons le plantage Mplayer avec les symboles de débogage :

```
$ gdb mplayer
Program received signal SIGSEGV, Segmentation fault.
(gdb) run run-0004/session-0004/flashmob-0001.mkv
0xb6f2419e in decode_packed_entry_number (book=0x8bd3a80, b=0x8bc75ec) at ../../
                                                        lib/codebook.c:315
315         long entry = book->dec_firsttable[lok];
(gdb) l
309     STIN long decode_packed_entry_number(codebook *book, oggpack_buffer *b){
310       int  read=book->dec_maxlength;
311       long lo,hi;
312       long lok = oggpack_look(b,book->dec_firsttablen);
313
314       if (lok >= 0) {
315         long entry = book->dec_firsttable[lok];
316         if(entry&0x80000000UL){
317           lo=(entry>>15)&0x7fff;
318           hi=book->used_entries-(entry&0x7fff);
319         }else{
(gdb) print book
$1 = (codebook *) 0x8bd3a80
(gdb) print book->dec_firsttable
$2 = (ogg_uint32_t *) 0x0
(gdb) print lok
$3 = 0
(gdb) print *book
$4 = {dim = 0, entries = 0, used_entries = 0, c = 0x0, valuelist = 0x0,
     codelist = 0x0, dec_index = 0x0, dec_codelengths = 0x0,
     dec_firsttable = 0x0, dec_firsttablen = 0, dec_maxlength = 0}
```

A priori, l'objet book n'est pas initialisé, car tous ses champs sont nuls. Pour contourner l'erreur (éviter le plantage), on peut remplace la ligne `long entry = book->dec_firsttable[lok];` par :

```
long entry;
if (book->entries <= lok) {
    return(-1);
}
entry = book->dec_firsttable[lok];
```

La valeur de retour `-1` est utilisée dans cette fonction pour indiquer une erreur. Une meilleure connaissance du projet permettrait de comprendre pourquoi `book` n'est pas initialisé et corriger le bug à la racine plutôt que de simplement le contourner.

Analyse des succès

On peut classer les succès d'un fuzzer en deux catégories :

- L'application quitte avec un code d'erreur ou est tuée par le système avec un signal (erreur de segmentation, instruction illégale, etc.).
- Dénis de service : l'application ne répond plus aux requêtes et parfois consomme l'ensemble des ressources du système (CPU et/ou mémoire).

Si la cible est un serveur, les deux catégories sont graves, car la conséquence est une interruption de service qui va bloquer l'ensemble des utilisateurs. D'ailleurs, dans le cas d'un déni de service, c'est souvent l'ensemble du système qui est affecté, les autres services sont également perturbés. Alors que si la cible est un logiciel client, comme un navigateur web ou bien un client de courriel, un déni de service n'aura qu'un faible impact.

Le cas le plus intéressant est celui de l'application tuée par un signal. En reproduisant l'erreur dans un débogueur, on peut obtenir plus d'informations sur l'état des registres, l'instruction ayant causé l'erreur, l'état de la pile, etc. Une erreur de segmentation

lors de l'appel à une fonction telle que `memcpy()` peut être due à un dépassement de tampon qui pourrait permettre de détourner le flot d'exécution. Néanmoins, avant de tirer la sonnette d'alarme, il faut vérifier que l'erreur est réellement exploitable.

Bien qu'un bug isolé puisse avoir un impact mineur, il se peut que plusieurs bugs utilisés simultanément permettent de monter une attaque complète aboutissant à une prise de contrôle de la cible. Le principe de précaution est donc de corriger tous les bugs.

Pour aller plus loin

Pour améliorer le fuzzer, on pourra lui greffer un débogueur qui permettra d'obtenir plus d'informations post-mortem et d'aider la détection de doublons. Un débogueur permet également d'avoir un meilleur aperçu du fonctionnement du programme, voire de le contrôler en cours d'exécution. Des fuzzers modifient la cible en cours d'exécution pour ignorer certaines fonctions comme la vérification d'une somme de contrôle.

Pour améliorer le fuzzer, on pourra lui greffer un débogueur...

On peut également tenter de rejouer la session d'un succès en diminuant progressivement l'agressivité pour réduire le « bruit » d'une attaque jusqu'à isoler les conditions minimales nécessaires pour reproduire le bug. Ceci facilitera d'autant l'isolation d'un bug.

Enfin, la fonctionnalité ultime d'un fuzzer serait la génération automatique d'un exploit démontrant la vulnérabilité. Plusieurs boîtes à outils proposent des exploits pour des failles connues, mais aucun fuzzer n'est capable de trouver comment utiliser le bug pour prendre le contrôle de la cible. Un exploit fonctionnel permet de gagner en crédibilité lorsqu'on demande à l'éditeur de corriger une vulnérabilité.

Conclusion

Cet article n'est pas une référence du fuzzing, mais présente simplement quelques points critiques lorsqu'on développe un fuzzer. Souvent, l'originalité est payante : rechercher de nouveaux vecteurs d'attaques ou de nouvelles manières de les exploiter permet de découvrir des portions de code moins testées et donc moins robustes. Les formats rares sont plus intéressants, car moins utilisés et donc moins bien testés.

La majorité des annonces de vulnérabilités actuelles sont issues du fuzzing, preuve que cette technique a encore de beaux jours devant elle.

Je vous conseille la lecture du livre **[Fuzzing]** qui présente plus en détail la technique du fuzzing sous Windows, Linux ainsi que Mac OS X.



Liens

[MILLER] MILLER (Barton P.), *Empirical Study of the Reliability of UNIX Utilities*, mars 1991. Copie disponible à l'adresse : **http://pages.cs.wisc.edu/~bart/fuzz/fuzz.html**

[SPIKE] Framework de fuzzing réseau écrit par Dave Aitel et Dug Song : **http://www.immunitysec.com/resources-freesoftware.shtml**

[sysfuzz] Fuzzer d'appel système (Linux et BSD) par la Digital Dwarf Society : **http://www.digitaldwarf.be/products/**

[Bunny] **http://code.google.com/p/bunny-the-fuzzer/**

[zzuf] **http://sam.zoy.org/zzuf/**

[Peach] **http://peachfuzz.sourceforge.net/**

[Fusil] Framework de fuzzing écrit en Python par Victor Stinner : **http://fusil.hachoir.org/**

[Valgrind] Suite d'outils de debug et de profiling de programme Linux : **http://www.valgrind.org/**

[Flayer] **http://code.google.com/p/flayer/**

[Fuzzing] SUTTON (Michael), GREENE (Adam) et AMINI (Pedram), *Brute Force Vulnerability Discovery*, édition Addison-Wesley, 2007.

Fabrice Flauss, Ingénieur d'études réseaux et télécoms – Division du système d'information
Département des infrastructures techniques – Rectorat de Nancy-Metz – Fabrice.Flauss@ac-nancy-metz.fr

RÉPARTITION DE CHARGES PAR LA PRATIQUE (PARTIE 1)

La mise en service de boîtiers de répartition de charges commence par l'apprentissage d'un nouveau vocabulaire dédié aux couches protocolaires hautes.

Cet article, en trois parties, explique ce vocabulaire, les concepts ainsi qu'une mise en œuvre alliant tolérance aux pannes matérielles, haute disponibilité, supervision et sécurité.

mots clés : ***haute disponibilité / services / contenus / keepalive***

1. Préambule

Lors d'une réflexion sur la sécurité des infrastructures techniques, une des questions porte parfois sur la mise en œuvre des boîtiers de répartition de charges pour améliorer la disponibilité des fermes de services, la tolérance aux pannes, leur supervision ainsi que leur gestion de contenus chiffrés. Ces différents aspects ont été présentés dans l'article précédent, paru dans *MISC* numéro 33 intitulé : « *répartition de charges : impacts potentiels sur la sécurité* ». Pour mettre en œuvre de la répartition de charges, on pense bien faire en appliquant quelques paramètres vérifiant la viabilité, le temps de réponse, la disponibilité, les capacités de traitement. Néanmoins, cela peut provoquer des dénis de services involontaires lors de montées en charge subites ou tout simplement par chevauchement de l'ensemble de ces paramètres. Dans cet article, nous verrons comment mettre tout cela en place.

Cet article a pris délibérément le choix de boîtiers dits « obsolètes » (*end of life*) afin de mieux expliquer l'incidence de moteurs SSL (*Secure Socket Layer*) externes, dont la configuration découle.

Il sera fondé sur l'architecture suivante :

- 2 répartiteurs CISCO CSS1150X ;
- 2 moteurs SSL SCA2-11000.

Le mode routé a été choisi, a contrario du mode *bridge*, afin d'expliquer la mise en application du VRRP (*Virtual router redundancy protocol*), les services dits « critiques », les services implicites et les paramètres implicites. En mode routé, le boîtier de répartition de charges contrôle la ferme de services : il en devient la passerelle par défaut. Il assure, de fait, le routage et le filtrage interzones des fermes de services.

Il s'agit d'appliquer les bonnes pratiques (*best practices*) de configuration de boîtiers de répartition de charges.

2. Présentation (Maquette simplifiée)

Cette maquette (cf. Figure 1), afin de débuter, n'utilise qu'un seul boîtier de répartition de charges et un seul moteur SSL. Au fur et à mesure de la mise en œuvre des configurations, nous ajouterons quelques strates à cette infrastructure, à des fins de stabilisation et de performances.

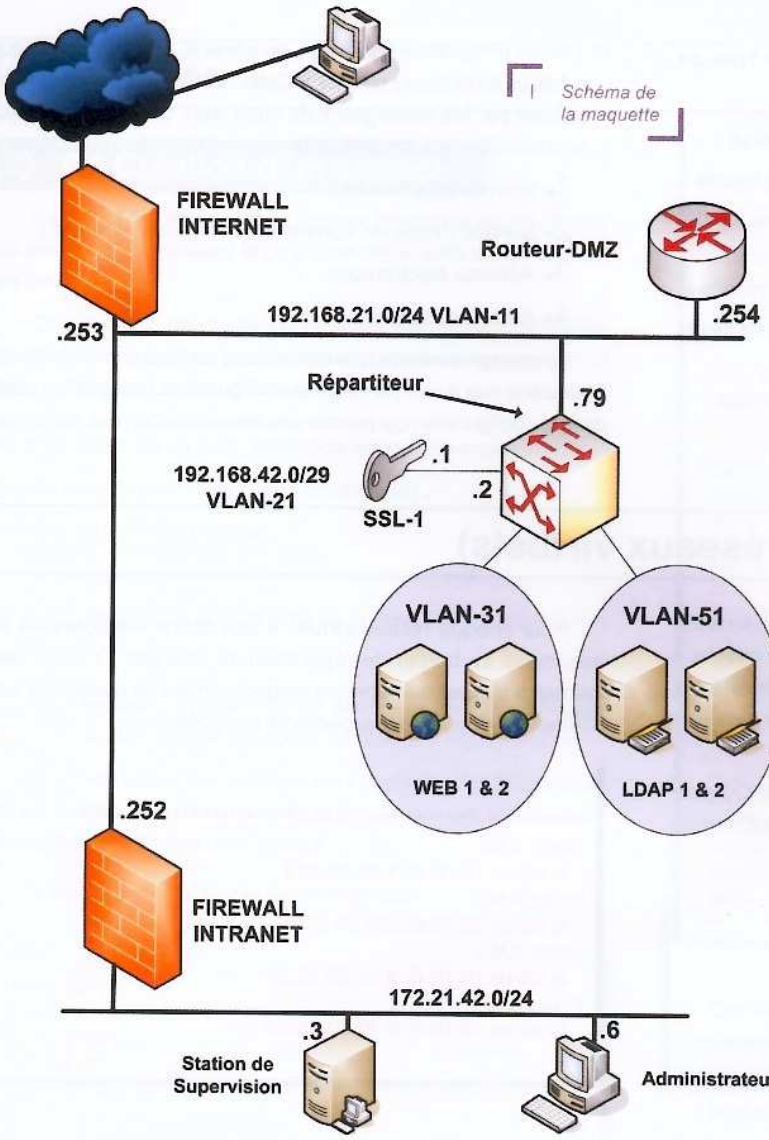


Schéma de la maquette

La figure 1 présente l'architecture qui servira d'exemple, avec la topologie suivante :

- une DMZ hébergeant les boîtiers de répartition de charges ayant comme *subnet* 192.168.21.0/24 (Réseau virtuel VLAN-11) ;
- un boîtier de répartition de charges ayant comme adresse 192.168.21.79/24 ;
- un moteur SSL1 ayant comme adresse IP 192.168.42.1/29 (Réseau virtuel VLAN-21) ;
- un *firewall* vers Internet ayant pour adresse IP 192.168.21.253/24 ;
- un routeur en DMZ ayant pour adresse IP 192.168.21.254 ;
- un firewall vers la zone INTRANET ayant pour adresse 192.168.21.252/24 ;
- l'ensemble des adresses IP virtuelles présentées par le boîtier de 192.168.21.20 à 192.168.21.25 ;
- le réseau INTRANET est alors adressé en 172.21.42.0/24, un serveur de supervision en 172.21.42.3, et par exemple un utilisateur en 172.212.42.6 ;
- les fermes de services sont adressées en 192.168.62.16/28 pour les webs (Réseau virtuel VLAN-31), pour les annuaires LDAP, on choisit le subnet supérieur soit le 192.168.62.32/28 (Réseau virtuel VLAN-51) ;

3. Vocabulaire

En préambule, il est fait état de l'apprentissage d'un nouveau vocabulaire, lors de la mise en œuvre de boîtiers de répartition de charges. Une explication de ces termes permet une meilleure compréhension des différentes lignes de configuration qui agrémentent cet article.

Le boîtier de répartition de charges ou communément appelé « CSS » (*Content Services Switch*), au sein de la maquette, architecture toute sa configuration autour de termes génériques définis ci-après :

- *Service* (Service) : un service est un objet représentatif de l'infrastructure technique. Il peut être un équipement de type routeur, firewall, un serveur appartenant à la ferme de services, un moteur SSL ou bien par exemple un cache, etc.
- *Content* (Contenu) : un contenu est la mise à disposition d'un service, tel que la publication de pages webs ou bien d'autres services tels que LDAP, FTP, DNS etc. On définit au sein du contenu l'URL d'accès, voire un port UDP ou TCP à disponibilité du client. On définit les services/serveurs qui possèdent ce contenu. Le boîtier répartira la charge en fonction des requêtes reçues et de la charge des serveurs cibles.

1. Lorsque la requête d'un client à destination d'un contenu est établie, le boîtier sait à quel service ou serveur s'adresser.
2. Ensuite, il aiguille le flux à destination du service ou serveur le plus approprié, à fournir, la mise à disposition du contenu.

3▸ La gestion du contenu permet, de définir le type de répartition de charges à appliquer :

- ⇨ *least connections* : sélection du serveur le moins utilisé ;
- ⇨ *round 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 ;
- ⇨ *sticky connections* : connexions liées en fonction de l'adresse IP source et/ou de l'adresse IP destination ;
- ⇨ *http header load balancing* : en fonction de l'en-tête http ;
- ⇨ *cookie* : cf. cookie HTML.

- ⇨ *Owner* (Propriétaire) : il s'agit, en général, de la personne qui publie un ou plusieurs contenus. Cette définition est généralement utilisée par les hébergeurs de sites web. Des champs dans la définition du propriétaire permettent de renseigner :

1▸ Nom du propriétaire ;

2▸ Service (financier, commercial, administration, etc.) ;

3▸ Adresse électronique ;

4▸ Adresse postale.

Le champ *owner* n'a que des notions de type administratives. Il n'accède pas à une partie de la configuration. Il s'agit d'un objet dans la configuration qui permet une hiérarchisation des contenus.

4. Définition des circuits (réseaux virtuels)

Le boîtier de répartition de charges possède des interfaces Ethernet appelées E1 à E9. Cette maquette utilise l'interface E9, qui servira également de lien 802.1Q (transport de réseaux virtuels) à destination du réseau d'accès (DMZ), des serveurs réels et du moteur SSL.

L'acronyme « *trunk* » est utilisé par le constructeur CISCO pour désigner un lien 802.1Q et non pour l'agrégation de liens Ethernet.

La configuration est alors celle ci-dessous :

```
!************************ INTERFACE ************************
interface e9
  trunk
  vlan 11
  vlan 21
  vlan 31
  vlan 51
```

Pour chaque réseau virtuel, il faut définir les adresses IP associées au boîtier de répartition de charges : il s'agit des mêmes adresses qui serviront respectivement de passerelle aux différents serveurs et équipements associés.

```
!************************ CIRCUIT ************************
circuit VLAN11
  ip address 192.168.21.79 255.255.255.0
circuit VLAN21
  ip address 192.168.42.2 255.255.255.248
circuit VLAN31
  ip address 192.168.62.30 255.255.255.240
circuit VLAN51
  ip address 192.168.62.46 255.255.255.240
```

5. Table de routage

La table de routage du boîtier de répartition de charges est très simple : une route par défaut vers chaque passerelle, dont une vers le moteur SSL, soit la configuration suivante :

`ip route 0.0.0.0 0.0.0.0 192.168.21.253 1` ← route par défaut vers le firewall accédant à l'Internet

`ip route 0.0.0.0 0.0.0.0 192.168.42.1 1` ← route par défaut vers le moteur SSL-1

`ip route 0.0.0.0 0.0.0.0 192.168.21.254 1` ← route par défaut vers le routeur de DMZ

`ip route 0.0.0.0 0.0.0.0 192.168.21.252 1` ← route par défaut vers le firewall accédant à l'intranet

Il reste néanmoins un problème quand on se situe en INTRANET au niveau du poste de l'administrateur, pour se connecter sur le boîtier : le constat est que cela ne fonctionne pas. En effet, il considère que c'est une attaque de type DoS : il est configuré pour fournir des services qui le traversent, et non sur lui-même.

Afin de résoudre ce problème de supervision et d'accessibilité à la configuration du boîtier sans passer par une interface dédiée au management, il y a deux approches : soit une route spécifique vers le réseau de supervision (dans ce cas, l'INTRANET), soit une route ayant le mot-clé « *originated-packets* ».

Ce réseau sera alors autorisé à effectuer des requêtes, par exemple, de types telnet, icmp, snmp, etc. sur le boîtier de répartition de charges.

Il est conseillé d'utiliser la seconde option qui filtre sur l'adresse IP ou réseau source.

Soit :

```
ip route 172.21.42.0 255.255.255.0 192.168.21.252 originated-packets
```

6. Services

6.1 Création d'un service

Il s'agit ici de définir chaque serveur faisant partie des fermes de services, en précisant le protocole utilisé ainsi que le numéro de port associé.

On y associe des batteries de tests, permettant de vérifier la viabilité du service. Ces tests sont appelés « *keepalive* » et peuvent être de natures différentes :

- *icmp* (tests via un *ping*, challenge echo/echo-reply);
- *http* (requête vers une URL par exemple) ;
- *none* (aucun test effectué) ;
- *tcp/port* (vérification de l'écoute d'un protocole) ;
- *script* (vérification approfondie de la mise à disposition d'une application).

Pour les tests de type http, on peut associer les méthodes `head`, `get` ou bien encore les codes de réponse http souhaités.

Pour l'ensemble des tests, on peut appliquer des fréquences de vérification, le nombre de tests, et le nombre maximal d'erreurs acceptées avant mise hors service.

Voici un exemple simple de configuration :

```
service WEB1
  ip address 192.168.62.21
  protocol tcp
  port 80
  keepalive type tcp
  keepalive port 80
  active
!
service WEB2
  ip address 192.168.62.22
  protocol tcp
  port 80
  keepalive port 80
  keepalive type tcp
  active
```

On va, dans ce cas, vérifier l'écoute (mode *listen*) du port tcp/80 des serveurs WEB1 et WEB2. On peut affiner, pour vérifier la mise à disposition de l'information sur le serveur en configurant un test de type : `keepalive type http` et `keepalive uri "/index.html"`.

Toutefois, les serveurs WEB1 et WEB2 peuvent héberger différentes instances APACHE, par exemple, ayant des contenus différents, un accès vers des serveurs d'applications autres, etc. Il est souhaitable alors de décliner chaque service, ayant une même adresse IP, mais ayant des contenus différents, comme dans l'exemple suivant :

```
service WEB1-photos
  ip address 192.168.62.21
  protocol tcp
  port 80
  keepalive type http non-persistent
  keepalive uri "/photos/test"
  active
!
service WEB2-photos
  ip address 192.168.62.22
  protocol tcp
  port 80
  keepalive type http non-persistent
  keepalive uri "/photos/test"
  active
!
service WEB1-videos
  ip address 192.168.62.21
  protocol tcp
  port 80
  keepalive type http non-persistent
  keepalive uri "/videos/test"
  active
!
service WEB2-videos
  ip address 192.168.62.22
  protocol tcp
  port 80
  keepalive type http non-persistent
  keepalive uri "/videos/test"
  active
```

Ces tests peuvent paraître simples, mais deviennent vite plus complexes dans le cas d'applications autres qu'un protocole natif tel que http.

Par exemple, pour des serveurs d'annuaires tels que le protocole LDAP, il ne s'agit pas de simplement tester l'écoute du port tcp/389, mais il peut être intéressant d'effectuer un *bind* sur le serveur via un script par exemple. Dans ce cas, un `keepalive type script` est approprié.

note

Ces services dits « implicites » ne sont vus ni en `show running`, ni en `show config`. Ils n'apparaissent que via la commande `sh keepalive`.

6.2 Services implicites

Pour chaque passerelle par défaut définie dans la table de routage, le boîtier de répartition de charges crée automatiquement un service. Ils sont visualisés de la manière suivante :

```
CSS11501-1# sh keepalive
Keepalives:

  Name: AUTO_nexthop00001  Index: 0  State: Alive
  Description: Auto generated for service nexthop00001
  Address: 192.168.21.253 Port: Any
  Type:            ICMP
  Frequency:        5
  Max Failures:     3
  Retry Frequency:  5
  Dependent Services:
    nexthop00001
```

Il faut faire très attention à ce type de service implicite qui teste obligatoirement par défaut la viabilité des divers objets via un test de type ICMP (ping). Il se peut que, suivant l'infrastructure, tel ou tel équipement n'accepte pas ce type de requêtes (par exemple, un firewall).

Une route par défaut mise à la main en mode configuration n'existera que si les tests fonctionnent. Il existe trois solutions à ce problème :

- La première consiste à autoriser les flux icmp depuis le boîtier de répartition de charges à destination de l'équipement visé (routeurs filtrant, firewalls, etc.)
- La deuxième consiste à interdire la création des services implicites **avant** de saisir la table de routage, grâce à la commande suivante :

```
no implicit-service
```

- Ou bien, si l'on souhaite déclarer les objets à des fins de supervision (cette déclaration annule le service implicite créé préalablement automatiquement), par exemple, on peut les déclarer comme suit :

```
service routeur-internet
  type transparent-cache
  ip address 192.168.21.253
  keepalive type none
  active
```

7. Caractéristiques des serveurs

On a vu précédemment comment mettre en œuvre des batteries de tests à des fins de vérification de la viabilité d'un service. Ils ont pour but de déterminer si le statut d'un service est disponible (Alive) ou bien hors service (Down).

Les serveurs qui composent une même ferme de services sont bien souvent hétérogènes, et il se peut que le statut disponible ne soit pas très fiable.

Alors, qu'en est-il de la charge des serveurs à un instant t ? Comment le boîtier de répartition de charges décide-t-il vers lequel des serveurs composant la ferme de services, il doit envoyer les requêtes d'un client ?

Prenons l'exemple du serveur B composant la ferme de services (serveurs A, B et C) dont on examine le statut :

```
CSS11501-1# sh service B

 Name: B          Index: 23
  Type: Local           State: Alive
  Rule ( 192.168.62.21  TCP  80 )
  Session Redundancy: Disabled
  Redirect Domain:
  Redirect String: (null)
  Keepalive: (HTTP:HEAD:/videos/test   5   3   5 )
  Last Clearing of Stats Counters: 01/01/2000 16:30:28

  Mtu:                      1500      State Transitions:          12
  Total Local Connections:  267657    Total Backup Connections:   0
  Current Local Connections: 0        Current Backup Connections: 0
  Total Connections:        267657    Max Connections:            65534
  Total Reused Conns:       100438
  Weight:                   1         Load:                       102
  DFP:                      Disable
```

Certains paramètres sont particulièrement intéressants :

- le statut du service est disponible : Alive ;
- les paramètres de tests sur l'URL **/videos/test** sont pré-paramétrés à : 5 3 5, soit :

```
Frequency:        5 ← test toutes les 5 secondes
Max Failures:     3 ← 3 tests négatifs maximum consécutifs
Retry Frequency:  5 ← temps de reprise après un échec soit
                      5 secondes
```

Il peut être intéressant de modifier certains paramètres par défaut, tels que la fréquence des keepalives. Mais, il faut utiliser ces paramètres avec prudence. Les tests de keepalive sont séquentiellement exécutés et peuvent par mégarde ou par diminution de leur itération, provoquer un déni de service involontaire. La fréquence par défaut est de 5 secondes, mais,

lors d'une charge subite et de l'utilisation par exemple d'un script pour vérifier la viabilité du service, le service peut reporter une valeur en échec ;

- *State transition* : le service B a changé d'état 12 fois depuis le 01/01/2008 ;
- *Load* : le service B a une valeur de load à 102 ? Il 'agit de l'estimation de la charge.

Il existe plusieurs méthodes d'estimation de la charge des serveurs. Celles-ci sont le plus couramment utilisées :

- charge relative (méthode par défaut) : temps de réponse d'un serveur relatif au prorata du meilleur temps de réponse de l'un des autres serveurs composant la ferme de services ;
- charge absolue : temps de réponse réel d'un serveur.

7.1 Charge relative

Une ferme de services est composée de trois serveurs A, B et C. Ils ont chacun, respectivement, un temps de réponse à un instant t de 100 millisecondes, 1100 ms et 120 ms.

L'échelle de la valeur `load` est de 2 à un seuil de 254 :

- le statut disponible (alive) est attribué pour les valeurs de 2 à 253 ;
- une fois le seuil de 254 atteint, le serveur ne recevra plus aucune requête (état `dying`), dans l'attente qu'il retrouve une valeur éligible ;
- le statut arrêté (down) sera affecté au serveur ayant la valeur 255.

Il existe un paramètre appelé `load step` qui permet de définir un pas ou une échelle de temps, afin de classifier les différents temps de réponse des serveurs. La valeur par défaut est de 10 ms. Elle peut être modifiée au gré des besoins, en fonction de l'homogénéité ou non de la ferme de services.

Dans l'exemple de la figure 2, les pas ont été réglés respectivement à 10 et 100 ms.

L'estimation du serveur B dans le cas du pas paramétré en 10 ms est réalisée comme suit :

- Le serveur ayant le meilleur temps de réponse est le serveur A, soit 100 ms ; le boîtier lui affecte la valeur minimale soit 2.
- La différence de temps de réponse entre le serveur B et le serveur A, est de 1000 ms ; ce résultat divisé par le pas paramétré ici 10 ms, nous donne le résultat de 100.
- Il suffit d'ajouter ces deux valeurs, le serveur B aura alors la valeur de 102.

La même estimation dans le cas d'un pas paramétré à 100 ms nous donne alors la valeur 12.

7.2 Charge absolue

Dans ce cas, le boîtier de répartition de charges estime le temps de réponse d'un serveur sans se soucier des temps de réponse des autres serveurs composant la ferme de services. La valeur correspondante est fonction d'un paramètre appelé « sensibilité » (`load absolute sensitivity`).

Cette valeur est par défaut égale à 21. Elle peut-être comprise entre 1 et 22, sachant que la valeur 1 est la granularité la plus fine.

L'estimation de la charge est répartie en 16 catégories maximales. En appliquant une sensibilité de 21 (celle par défaut), cela donne le tableau 1, page suivante.

2 *Charge relative*

Nom du serveur	Temps de réponse
Serveur A	100 ms
Serveur B	1100 ms
Serveur C	120 ms



Tableau 1 : Charge absolue avec une sensibilité par défaut de 21

Catégories	Valeur de la charge	Pas en ms	Temps de réponse maximum (ms)	h:mn:s
1	2-15	2	32	0 :0 :0
2	16-31	4	96	0 :0 :0
3	32-47	8	224	0 :0 :0
4	48-63	16	480	0 :0 :0
5	64-79	32	992	0 :0 :0
6	80-95	64	2016	0 :0 :2
7	96-111	128	4064	0 :0 :4
8	112-127	256	8160	0 :0 :8
9	128-143	512	16352	0 :0 :16
10	144-159	1024	35736	0 :0 :32
11	160-175	2048	65504	0 :1 :5
12	176-191	4096	131040	0 :2 :11
13	192-207	8192	262112	0 :4 :22
14	208-223	16384	524256	0 :8 :44
15	224-239	32768	1048544	0 :17 :28
16	240-254	65536	2031584	0 :33 :31

Tableau 2 : Charge absolue avec une sensibilité de 1

Catégories	Valeur de la charge	Pas en ms	Temps de réponse maximum (ms)	h:mn:s
1	2-15	1	16	0 :0 :0
2	16-31	1	32	0 :0 :0
3	32-47	1	48	0 :0 :0
4	48-63	1	64	0 :0 :0
5	64-79	1	80	0 :0 :0
6	80-95	1	96	0 :0 :0
7	96-111	1	112	0 :0 :0
8	112-127	1	128	0 :0 :0
9	128-143	1	144	0 :0 :0
10	144-159	1	160	0 :0 :0
11	160-175	1	176	0 :0 :0
12	176-191	1	192	0 :0 :0
13	192-207	1	208	0 :0 :0
14	208-223	1	224	0 :0 :0
15	224-239	1	240	0 :0 :0
16	240-254	1	255	0 :0 :0

Si l'on remarque les temps de réponse attendus, dans le tableau 1, avant de sortir un serveur de la ferme de services, il se peut, ce qui arrive dans la plupart des cas, qu'il soit inacceptable de laisser la sensibilité par défaut. Un client ne comprendrait pas devoir attendre 20 ou 30 minutes pour recevoir une page Web par exemple.

Cet exemple typique, oblige l'administrateur de boîtiers de répartition de charges, à la modification du paramètre `load-absolute-sentivity`, afin que le service rendu à l'utilisateur soit le plus proche possible du résultat escompté.

Si l'on modifie à la valeur la plus fine soit la valeur 1, le tableau 2 présente alors les nouvelles valeurs.

7.3 Charge relative ou charge absolue ?

Dans la plupart des cas, la charge dite « relative » est la plus appropriée, car si la ferme de services comporte des serveurs de capacités de traitements différentes, ils seront comparés les uns aux autres en temps réels, afin de répondre efficacement à la requête d'un client.

Mais, en cas d'homogénéité d'une ferme de services, la charge dite « absolue », avec un réglage fin de la granularité permet de répondre à un excellent niveau de qualité de services.

note

Dans l'exemple du tableau 2, on remarque que le pas est alors linéaire.

7.4 Mode maintenance

Dans le cadre d'une opération de maintenance sur un serveur, telle que le remplacement de barrettes mémoires, un ajout de mémoire, la modification d'un système de fichiers, une mise à jour de la publication de pages web par exemple ou bien encore une migration vers un nouveau serveur, il est souhaitable de répondre à une qualité de services irréprochable vis-à-vis de la requête d'un client. On ne peut plus interagir via le serveur réel, c'est au boîtier de répartition de charges de mettre hors service le serveur réel, en continuant à écluser les sessions actives en cours, et en arrêtant les nouvelles connexions entrantes à destination de celui-ci.

La manière de procéder est d'agir sur le paramètre poids (*Weight*) :

```
CSS11501(config-WEB1-videos)# Weight 0
```

Alors, le boîtier de répartition de charges attendra que les sessions en cours se terminent et le service WEB1-videos sera arrêté.

D'autre part, tout le monde s'est retrouvé devant une page dite « en mode maintenance » ou disant « veuillez bien nous excuser, une opération de maintenance est en cours ». Comment arrivent-ils à faire cela et de quelle manière ? Un boîtier de répartition de charges peut vous apporter son aide.

En fait, il est possible de créer un service spécifique, qui par la suite sera associé à un contenu (cf. chapitre 8). On l'appelle dans notre exemple `maintenance`. Il se définit comme suit :

```
service maintenance
  ip address 192.168.62.25
  protocol tcp
  port 80
  keepalive type http non-persistent
  keepalive uri "/sorry.html"
  active
```

Le serveur dit `maintenance` est pré-paramétré pour accepter tous types de requêtes. Il affichera alors au client ayant effectuée sa requête la phrase souhaitée.

Dans le chapitre 8, des exemples seront présentés avec l'utilisation de ce service.

8. Gestion des propriétaires et des contenus

8.1 Propriétaire

On catégorise des contenus à des fins de clarté de lisibilité de configuration.

On peut créer ainsi des propriétaires de type :

```
OWNER HTTP-80
OWNER HTTP-81
OWNET LDAP
OWNER SSL
```

Dans cette classification, on met les contenus à visibilité en protocole natif tel que http et ldap au sein des propriétaires respectifs HTTP-80 et LDAP.

Pour le protocole à visibilité SSL, un propriétaire du même nom lui est désigné. Les contenus classifiés au sein du propriétaire HTTP-81 seront eux réservés à la visibilité du/des moteur(s) SSL.

8.2 Contenus

On associe les divers serveurs à notre disposition aux contenus. On leur applique une méthode de répartition de charges adéquate. Les méthodes les plus utilisées sont généralement les plus simples :

- S'il s'agit d'un contenu statique, on utilisera `balance leastconn` → envoi vers le serveur le plus disponible.
- S'il s'agit d'un contenu dynamique, on utilisera une méthode avancée `advanced balance sticky srcip` → maintien de la connexion liée par adresse IP source.

On souhaite positionner une adresse IP virtuelle 192.168.21.20 (visibilité DMZ) pour accueillir les fermes de services, fournissant les contenus en http (port 80) `/photos` et `/videos`. Cela s'écrit comme suit :

```
!************************** OWNER ****************************
owner HTTP

  content PHOTOS
    vip address 192.168.21.20
    balance leastconn
    protocol tcp
    port 80
    url "/photo*"
    add service WEB1-photos
    add service WEB2-photos
    active

  content VIDEOS
    vip address 192.168.21.20
    advanced-balance sticky-srcip
    protocol tcp
    port 80
    url "/video*"
    add service WEB1-videos
    add service WEB2-videos
    sticky-inact-timeout 15 ← 15 minutes
    active
```

En mode `advanced-balance sticky-srcip`, la table de flux est conservée pendant la valeur du *timer*. Il se peut que, lors d'une supervision, on se rende compte que cette valeur était inadéquate avec la qualité de services escomptée, et qu'un seul serveur de la ferme de services était le plus souvent sollicité. Il faut alors procéder comme suit :

```
CSS11501-1# llama ← accès au mode debug

CSS11501-1(debug)# sticky-purge all-sticky
```

Cela purgera la table des adresses IP sources ; le seul moyen de le faire est l'accès au mode debug.

Le masque réseau par défaut de l'adresse IP source est 255.255.255.255 : il est possible de le modifier via la commande

`sticky mask`, afin de résoudre des problématiques de type proxy provenant de fournisseurs d'accès.

Dans le cadre d'une maintenance sur l'un des serveurs, voire de leur globalité, leur nombre n'étant pas limité, on peut associer un serveur dit « de maintenance », tel qu'on l'a vu précédemment lors du chapitre 7.4.

Ce service spécifique se charge de relayer l'ensemble des serveurs composant la mise à disposition dudit contenu :

```
content VIDEOS
  vip address 192.168.21.20
  advanced-balance sticky-srcip
  protocol tcp
  port 80
  url "/video*"
  add service WEB1-videos
  add service WEB2-videos
  primarySorryServer maintenance  ← affectation du service maintenance
  sticky-inact-timeout 15
  active
```

Lorsque WEB1-videos et WEB2-videos seront arrêtés, le serveur de maintenance prendra le relais. On arrête ainsi la publication d'un contenu sans que le client ait un beau HTTP/404, mais plutôt un message approprié.

8.3 Redirection d'URL

On peut rediriger un client ayant par mégarde utilisé une URL obsolète. Il s'agira d'un contenu spécifique :

```
content MONDOMAINE
    vip address 192.168.21.20
    url "//photos.mondomaine.org/*"
    protocol tcp
    port 80
    redirect "http://www.mondomaine.org/photos/"
    active
```

9. La suite au prochain numéro…

On se doit d'apporter aux clients des services irréprochables ou tendant à l'être. Il faut, pour ce faire, veiller à une bonne homogénéité de la qualité de service des différentes strates au sein des infrastructures techniques.

Il nous reste quelques étapes avant de stabiliser et d'offrir ce service irréprochable : la sécurisation des transactions, la mise en œuvre de la tolérance aux pannes matérielles des boîtiers de répartition et des moteurs SSL, la mise en œuvre d'un système de supervision des fermes de services et des boîtiers, des graphiques associés. Ensuite, il faudra se focaliser sur la sécurité des fermes de services, les règles de filtrage interzones, etc.

Ce sera lors d'un prochain numéro…

Remerciements

Je souhaite remercier Sarah NATAF et Fred RAYNAL, pour leurs relectures successives et conseils avisés.

Bibliographie

- Point d'entrée global sur la documentation du boîtier de répartition de charges : **http://www.cisco.com/univercd/cc/td/doc/product/webscale/css/css_820/**
- Vocabulaire : **http://www.cisco.com/en/US/docs/app_ntwk_services/data_center_app_services/css11500series/v8.20/configuration/content_lb/guide/Overview.html**
- Routage : **http://www.cisco.com/en/US/docs/app_ntwk_services/data_center_app_services/css11500series/v8.20/configuration/routing/guide/Intface.html**
- Services : **http://www.cisco.com/en/US/docs/app_ntwk_services/data_center_app_services/css11500series/v8.20/configuration/content_lb/guide/Services.html**
- Gestion des scripts : **http://www.cisco.com/en/US/docs/app_ntwk_services/data_center_app_services/css11500series/v8.20/configuration/administration/guide/Scripts.html**
- Charge relative ou charge absolue : **http://www.cisco.com/en/US/docs/app_ntwk_services/data_center_app_services/css11500series/v8.20/configuration/content_lb/guide/Load.html**
- Gestion des propriétaires et des contenus : **http://www.cisco.com/en/US/docs/app_ntwk_services/data_center_app_services/css11500series/v8.20/configuration/content_lb/guide/ContRule.html**