misc
35 - Janvier/Février 2008

misc

Multi-System & Internet Security Cookbook

35 Janvier Février 2008

100 % SÉCURITÉ INFORMATIQUE

[DOSSIER]

AUTOPSIE & FORENSIC

COMMENT RÉAGIR APRÈS UN INCIDENT ?

Analyse post mortem, récupération de données, e-discovery, réponse à incident, forensique

guérilla / guerre / protection / architecture / vulnérabilités

GUERRE DE L'INFO

Du trafic d'armes numériques à la protection des infrastructures (p. 04)

[SYSTÈME]

Comprenez les systèmes de sécurité d'OpenSolaris et Solaris (p. 50) *durcissement / système / configuration / sécurisation*

[RÉSEAU]

Supervision et sécurité par analyse de flux (p. 62) *représentation / trafic / attaques / détection d'intrusion / métrologie*

FICHE TECHNIQUE

Reverse proxy Apache 2.2 : le couteau suisse sécurité de vos applications (p. 70) *reverse proxy / sécurisation transparente pour l'application / Apache*

[Édito]

Soldes divers(e)s

S'occuper d'une revue n'est pas toujours une sinécure, en particulier en période de Noël : on mange trop et on pense au cadeau envoyé par tata Rodriguez directement depuis le Portugal en paquet Fado (Desproges). Avec ça, essayez d'écrire un édito digne de ce nom, et de prendre un peu de recul sur six années d'existence...

En fait, je traîne dans ce milieu depuis une dizaine d'années, ce qui n'est rien comparé à certains respectables dinosaures. Mais, quand je regarde dans le rétroviseur, je me dis que, finalement, ça ne change pas beaucoup. Si j'étais candidate à l'élection de Miss France, je voudrais la paix dans le monde et de quoi boire et manger pour tous les hommes. Comme je suis cynique et que je travaille dans un monde qui ne l'est pas moins, je dis « tant mieux » : si les choses ne changent pas, ça veut dire que je devrais avoir du boulot pour quelques décennies encore (et accessoirement mes *padawans* aussi, parfait pour mes cotisations retraite).

Mais, je m'égare (et avec mon sens de la poésie, c'est forcément mon parnasse). Je voulais traiter des phénomènes de mode. Pas la *techtonik*, les Pokémons et Lorie, ceux en sécurité. On a eu les *firewalls* (avec l'évolution révolutionnaire *stateful*), puis les IDS (avec l'option révolutionnaire « mode coupure », aussi appelée « IPS »). Tiens, c'est bizarre, à aucun moment on a eu de *buzz* sur les anti-virus. Comme quoi, ça ne doit fondamentalement pas servir à grand chose ces bêtes-là. Mais, à côté des évolutions technologiques, la manière de faire de la sécurité, d'en parler ou de la vendre a également changé.

Fini le petit artisan qui travaille avec ses outils et ses mains. Plus le temps pour ça, on est entré dans l'ère industrielle. Maintenant, il faut un processus sécurité validé par la qualité : les méthodologies et normes diverses et variées fleurissent partout : ISO17799 devenue ISO27002, COBIT/ITIL ou encore les certifications CISSP, GIAC, CISA, etc. Les cabinets se font certifier (puis deviennent eux-mêmes organismes formateurs). Phénomène amusant, quand l'un est certifié, le voisin se dit qu'il va devoir en faire autant, ce qui engendre une course aux certifications. Concrètement, ça épuise les ressources des concurrents, et place le premier organisme formateur en situation de leader : joli coup.

Quant au discours autour de la sécurité, un mot clé apparaît de plus en plus : intelligence économique (IE). Bien sûr, ce domaine est largement plus vaste que simplement la sécurité informatique. Une amie le définit comme du renseignement légal, j'aime bien, même si ça réduit à la collecte et à l'analyse d'informations. Bref, toujours est-il que de plus en plus de boîtes de sécurité informatique « vendent » de l'IE ou, inversement, des cabinets d'IE vendent de la sécurité informatique. En conséquence, le discours évolue pour faire craindre à tout va la stagiaire chinoise ou le mafieux russe qui s'en prendra à vos biens. On joue beaucoup sur la peur et le fantasme excitant de l'espion, et ça marche.

Ce qui me titille derrière l'oreille avec tout ça, c'est ce que dissimulent ces belles paroles (normes ou IE). D'abord, il est intéressant de constater comme tout cela s'emboîte bien : on distille des craintes, puis on se rassure en appliquant des méthodologies pas à pas. Mais, c'est surtout oublier que les logiciels ont toujours autant de failles, que les firewalls ne servent presque plus à rien, puisque tout passe par les ports 80/443, et que les services sécurité n'ont toujours pas les moyens de remplir leurs missions. C'est très semblable à un politique qui commence par faire peur, puis tient de beaux discours en faisant de belles promesses, mais ensuite, une fois aux affaires, n'a pas les moyens de les mettre en œuvre.

Ne vous méprenez pas, je ne dis pas que tout cela est inutile, mais il faut arrêter de croire que la Solution est là (oui, oui, celle avec un S majuscule qui va tout régler magiquement). Par exemple, le déroulement d'une méthodologie, c'est aussi du pain béni pour l'attaquant, puisque ça lui balise tout le chemin suivi par le défenseur. J'ai peut-être le tort de croire que la créativité et la rigueur sont des qualités essentielles pour traiter ces problèmes anormaux : halte à l'économie d'intelligence.

Il y a quelques années, cela m'aurait énervé. Maintenant... ben, c'est pareil :-) Mais, avec l'âge (pardon, l'expérience) et le recul (pardon, l'expérience encore), je relativise. Quelque part, savoir que MISC entame encore une nouvelle année me rassure. Cela signifie que des personnes (lecteurs, auteurs, ma grand-mère et mes neveux) sont encore intéressées par mettre les mains dans le cambouis et affronter les difficultés où elles se trouvent. Merci à tous de votre soutien, de votre participation, de vos remarques, et surtout bonne année et bonne lecture !

Fred Raynal

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.

[Sommaire] 35

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

Directeur de publication : Arnaud Metzler

Chef des rédactions : Denis Bodor
Rédacteur en chef : Frédéric Raynal
Secrétaire de rédaction : Véronique Wilhelm
Relecture : Dominique Grosse
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

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

Guerre, guérillas et terrorismes informatiques : du trafic d'armes numériques à la protection des infrastructures

Dans cet article qui fait suite à celui publié dans un numéro précédent [18], nous abordons deux aspects importants dans tout conflit : celui de la protection des infrastructures critiques, lesquelles sont vitales pour un pays, et le problème du trafic des armes numériques, qui permettent à des attaquants sans grande compétence technique de mener des attaques à l'aide d'outils sophistiqués. De ce point de vue, le domaine du numérique n'est fondamentalement pas différent des contextes conventionnels. La seule différence tient au fait que si dans ce dernier la prise de conscience est aiguë, dans le domaine du numérique, le déficit de perception est encore immense. La raison essentielle tient au fait que la seule vision technique, voire technicienne de la lutte informatique offensive est prise en compte : le contexte opérationnel plus large est ignoré. La plupart des gens, et souvent les décideurs eux-mêmes, sont incapables de définir la dépendance réelle dans notre société des systèmes d'information et de mesurer l'impact qu'aurait effectivement une attaque informatique contre les infrastructures de leur pays. Enfin, cet article présente quelques compléments et mises à jour concernant le conflit russo-estonien, certains éléments ayant pu être obtenus de sources étrangères fiables [1] et ayant participé à la « gestion » de conflit.

mots clés : ***protection / architecture / préfixe / processeur***

Si le lecteur a vu le dernier volet de la tétralogie *Die Hard*, intitulé non sans à propos, dans sa version originale, *Die Hard 4.0*, alors ce dernier a pu avoir une idée assez précise, mais néanmoins globalement exacte – aux « fioritures hollywoodiennes » près – de la façon dont un conflit numérique – du simple attentat par des e-jihadistes à une véritable offensive étatique – pourrait être mené et les impacts qu'il pourrait vite avoir. Le grand chaos résultant d'attaques graduées, multi-niveaux contre les infrastructures d'un pays hyper dépendant des technologies de l'information n'est plus un fantasme cinématographique, mais bien une éventualité à laquelle il faut se préparer.

Distribution de ressources et d'énergie interrompue, hôpitaux bloqués, services de secours et d'alerte paralysés, réseau de communication inopérant… nous montrons que ces attaques, prises séparément ont déjà été rencontrées ou envisagées et simulées. À chaque fois, le constat est dramatique et montre le dénuement complet dans lequel se trouvent nos infrastructures. Si les experts sont convaincus de la faisabilité technique, les décideurs et les politiques sont encore à des lieues d'appréhender correctement les problèmes.

L'autre aspect intéressant du film, encore une fois replacé dans un contexte cohérent, est de présenter ce qui existe déjà : le recours à des cyber-mercenaires qui peuvent tout à la fois se révéler être également des trafiquants d'« armes numériques ». C'est également ce que nous expliquons ici. L'offre de service et de produits dans le domaine des attaques informatiques est là : elle est conséquente et elle est organisée.

Le conflit russo-estonien, présenté dans [18], pourrait n'être finalement qu'une répétition générale en vue du grand chaos auquel pourrait être confronté un pays. Les derniers éléments que nous avons pu obtenir d'une source étrangère compétente semblent le suggérer fortement.

1. Mercenaires et trafiquants d'armes numériques

Dans tout conflit ou toute guerre, quelle que soit sa nature ou son ampleur, deux types d'acteurs particuliers, parmi les acteurs classiques, interviennent et doivent être connus : les marchands d'armes et les mercenaires ou les soldats privés. Des conflits dits « de basse intensité [2] » comme l'Afghanistan ou l'Irak montrent d'ailleurs, en particulier, une évolution autant significative qu'inquiétante, au moins pour la seconde catégorie, dans ce domaine. Les états qui répugnent de plus en plus à s'engager – pour diverses raisons – y recourent. La situation n'est pas différente dans le domaine des attaques numériques : les cyber-mercenaires, les marchands de « canons numériques » et autres officines privées existent et semble-t-il jouent un rôle appelé à devenir sinon prépondérant, du moins important.

Dans cette partie, nous considérons ces deux types d'acteurs à travers quelques exemples, qui plus qu'un long exposé sur le sujet – si tant est qu'il soit réellement possible – en disent pas mal sur la réalité de la menace qu'elle représente. D'un point de vue général, nous distinguerons :

- Les « trafiquants d'armes numériques ». Il s'agit d'individus offrant contre rétribution des programmes ou données permettant de réaliser des attaques informatiques. Dans le cas des données, ils assurent également le rôle d'espions privés, puisqu'ils se substituent à leurs « clients » pour collecter des informations sur la cible.
- Les cyber-mercenaires qui eux vont, contre rétribution ou pour des motifs idéologiques, réaliser des attaques sur demande ou dans un esprit militant.

L'expérience montre que quelquefois ces deux catégories peuvent être confondues. Et la déclaration suivante, de Valérie McNiven [22],

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

encadré 1

Estonie – petit retour en arrière

Des informations récentes, obtenues de sources a priori « bien informées » (présentes en Estonie en avril et mai, elles ont participé activement à la défense des infrastructures estoniennes aux côtés du CERT-ee) permettent de préciser le périmètre de ces attaques.

L'essentiel des attaques a été menée à partir de *botnets*. Rien de bien surprenant lorsque l'on s'est penché sur les logiciels téléchargeables, ou les scripts échangés, et destinés à attaquer les sites estoniens. Leur efficacité n'est que très relative.

Trois botnets différents ont été utilisés lors de ces attaques. Chacun était dédié à un type d'attaque précis : un premier réalisait des attaques par ICMP *flooding*, un deuxième faisait du TCP SYN flooding et le troisième s'en prenait aux DNS. Ces trois réseaux rassemblaient moins de 1000 PC zombies. Ils semblent avoir mené leurs attaques pendant 10 jours pratiquement sans interruption avec une intensité plus ou moins violente.

La composition de ces botnets s'est révélée surprenante : alors que ces attaques sont généralement menées par des réseaux constitués d'ordinateurs se trouvant principalement dans les pays de l'Est et en Asie, il a été observé une proportion très faible pour ces pays (moins de 5 %) alors qu'ils en ont identifié environ 23 % des ordinateurs comme se trouvant aux États-Unis (2 % environ étaient attendus).

Se pose la question de l'origine ou de l'ordonnateur de ces attaques. Russie ? Pas Russie ? La question ne sera a priori jamais tranchée définitivement.

Si c'était le cas, le déplacement du monument à l'origine des incidents pourrait n'avoir été que le prétexte à un test dont les objectifs auraient été, principalement, d'évaluer la réfutabilité, la capacité à nier toute participation à ce genre d'attaque (de fait personne n'aura jamais de preuve de quoi que ce soit – Les Estoniens, qui étaient les premiers, en mai dernier, à accuser ouvertement Moscou d'être derrière ces attaques le reconnaissent maintenant) et la capacité de réaction des grandes organisations internationales (en l'occurrence, la Communauté européenne et l'OTAN) face à ce genre d'attaque.

conseiller en matière de cybercriminalité auprès du Trésor américain (équivalent de notre ministère de l'Économie) est éloquente :

« Depuis l'année dernière [2004], le 'chiffre d'affaire' de la cybercriminalité a dépassé celui du trafic de drogues illégales…la cybercriminalité progresse si vite qu'il est devenu quasi impossible pour la justice de l'endiguer ».

Plus que les réalisations d'attaques elles-mêmes, la part du trafic de données et de logiciels offensifs représente la plus grande partie de cette nouvelle « économie ».

1.1 Les trafiquants d'armes numériques

Acteurs incontournables de tout conflit, le domaine du numérique n'échappe pas à ce constat, bien au contraire. La première des raisons tient au fait que le nombre de clients potentiels est bien plus important, la seconde est qu'il est relativement facile de proposer des produits très sophistiqués et que les contrôles et régulations internationaux – contrairement à l'armement classique, le nucléaire ou le chimique – sont illusoires. Au fond, produire un virus ou un *rootkit* évolué, un ver espion, est à la portée de tout bon informaticien et nul inspecteur de l'ONU ou d'une autre agence ne pourra inspecter tous les sites de production potentiels.

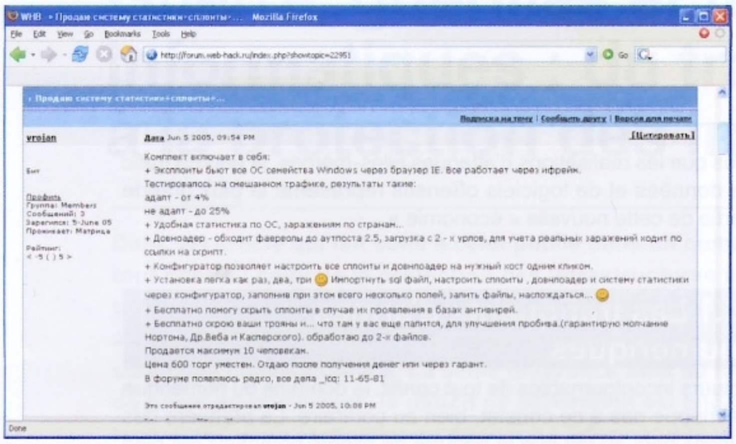
Le domaine du numérique a ceci de particulier qu'il est accessible au plus grand nombre, enseigné dans les écoles d'ingénieurs et les universités du monde entier. Autrement dit, le nombre potentiel de « marchands de canons numériques » est par définition très élevé.

1.1.1 Le trafic de logiciels d'attaques

La tendance n'est malheureusement pas nouvelle : le marché est juste passé du petit artisanat à une véritable industrie. Dès 1991, avec la mise sur le marché d'un logiciel comme *Virus Creation Lab*, il était possible pour toute personne ayant des volontés de nuisance nettement supérieures à sa maîtrise de l'informatique, de disposer d'un générateur automatique de virus qu'il suffisait ensuite d'utiliser. Depuis cette date, près de 500 générateurs de ce type, chaque fois plus évolués, ont été depuis répertoriés. D'autres outils ont également fait leur apparition dans le domaine des réseaux : générateurs automatiques de vers, outils de déni de service (voir par exemple le cas du logiciel Zyklon utilisé lors du conflit russo-estonien [18]), outils ou services d'anonymisation… bref, toute la panoplie est disponible à celui qui sait chercher. Une première révolution dans ce petit monde est survenue avec le commerce des exploits – autrement dit des vulnérabilités critiques de logiciels. L'année 2003, de ce point de vue semble représenter un tournant : l'attaque par le ver Blaster, rendue possible par des vulnérabilités du protocole RPC, ou, plus récemment, en janvier 2006, l'attaque contre le parlement britannique en exploitant la faille WMF ont ceci en commun que le ou les attaquants ont profité d'exploits vendus par certains groupes d'Europe de l'est.

De ce point de vue, ce marché des « exploits » est très florissant : une vulnérabilité critique se négocie entre 4000 et 50000 $. De quoi susciter bien des carrières de trafiquants ! Pour illustrer cela, le meilleur exemple est celui de l'annonce décrite en figure 1, page suivante.

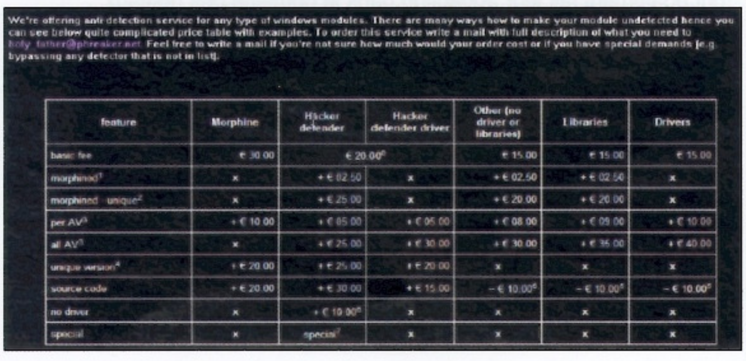
En juin 2005, dans un *post* sur le site **Web-hack.ru**, un dénommé « vrojan » offrait différents exploits pour la vulnérabilité IFRAME,



1 Offre de vente d'un exploit IFRAME sur le site Web-hack.ru

ainsi que des services de statistiques mesurant leur efficacité. L'auteur de ce post affirmait avoir testé ses exploits à la fois contre divers sites, avec chaque fois plusieurs niveaux de réponses pour chacun d'eux. Assurant que leur mise en œuvre se fait en « trois clicks » au moyen d'un simple fichier SQL contenant à la fois le code de l'exploit et l'outil de mesure statistique, l'auteur enfin annonce la capacité de rester indétecté par des antivirus comme Norton, DrWeb ou Kaspersky. Précisons que cette offre était limitée à 10 personnes ce qui laisse augurer un prix relativement élevé.

Ce marché des exploits, et en particulier celui des fameux 0-days [19], proposé sur des sites finalement accessibles – la maîtrise du russe, du chinois ou de langues tout aussi exotiques est indispensable – n'est que la partie émergée d'un iceberg dont il n'est probablement pas possible d'évaluer la taille exacte. Il existe très probablement un marché encore plus souterrain accessible uniquement à des gens très introduits, ne faisant pas de publicité sur Internet.



We're offering anti detection service for any type of windows modules. There are many ways how to make your module undetected hence you can see below quite complicated price table with examples. To order this service write a mail with full description of what you need to holy_father@phreaker.net Feel free to write a mail if you're not sure how much would your order cost or if you have special demands [e.g. bypassing any detector that is not in list].

feature	Morphine	Hacker defender	Hacker defender driver	Other (no driver or libraries)	Libraries	Drivers
basic fee	€ 30.00	€ 20.00[6]		€ 15.00	€ 15.00	€ 15.00
morphined[1]	x	+ € 02.50	x	+ € 02.50	+ € 02.50	x
morphined - unique[2]	x	+ € 25.00	x	+ € 20.00	+ € 20.00	x
per AV[3]	+ € 10.00	+ € 05.00	+ € 05.00	+ € 08.00	+ € 09.00	+ € 10.00
all AV[3]	x	+ € 25.00	+ € 30.00	+ € 30.00	+ € 35.00	+ € 40.00
unique version[4]	+ € 20.00	+ € 25.00	+ € 20.00	x	x	x
source code	+ € 20.00	+ € 30.00	+ € 15.00	– € 10.00[5]	– € 10.00[5]	– € 10.00[5]
no driver	x	+ € 10.00[6]	x	x	x	x
special	x	special[7]	x	x	x	x

2 Tarifs avec options du rootkit Hacker Defender

La seconde révolution est plus récente et date de fin 2005, début 2006. Elle concerne le marché des rootkits, ces kits de furtivité permettant de rendre invisible des programmes (que ce soit les fichiers eux-mêmes ou les processus associés) [3]. Là également, une offre diversifiée est disponible. Le cas du rootkit Hacker Defender est à ce propos particulièrement parlant. Proposé sur le site **http://hxdef.czweb.org**, il est disponible en différentes versions : bronze (150 euros), argent (250 euros), or (450 euros) et diamant (580 euros) et avec différents niveaux de prestations et technologies comme le montre la figure 2.

1.1.2 Le trafic de données

Il est encore plus important que celui présenté dans le paragraphe précédent. Ces données sont essentiellement de deux types :

- Des données traitant d'un système, dont la connaissance permet ensuite de mener une attaque contre le système cible, que ce soit au niveau de son utilisateur (chantage, pressions de toutes sortes…) ou le système qu'il sert. Il s'agit donc là du renseignement au sens classique du terme.
- Des données permettant de capter des fonds – le nerf de toute guerre – de manière frauduleuse soit dans un but purement crapuleux, soit pour financer d'autres attaques que ces attaques soient conventionnelles – préparation d'attentats par exemple – ou numériques – location d'un botnet ou de cyber-mercenaires. Le vol et le trafic de ces données peuvent suffire en soi à saper la confiance que ce soit dans une entreprise victime de ces vols ou dans un service tout entier, comme le commerce électronique. C'est finalement l'un des buts recherchés par le e-jihad [20].

Il est assez sidérant de constater combien la collecte de ces données se fait facilement. Le sentiment du grand public dans la sécurité des sociétés dont nous sommes les clients est proche de la naïveté et d'un manque certain de transparence de ces dites sociétés. Qui se douterait que beaucoup de sociétés – et la France n'est pas exempte de reproches – font souvent de l'externalisation des services de supervision de leur propres systèmes, certains critiques, par de sociétés basées à l'étranger et dont les canons en matière de sécurité et de recrutement de leurs personnels est loin de valoir les nôtres ? Dès lors, obtenir des informations sur une société, sur ses réseaux devient assez facile. Le souci de délocaliser afin de réduire les coûts se traduit par une diminution de la sécurité dans bien des cas, au moins potentiellement.

Il serait possible de citer des centaines d'exemples – et pour certains la discrétion est préférable. Citons-en cependant quelques-uns qui ont valeur de symboles :

- En décembre 2006, il était possible, sur le site **celltolls.com** d'acquérir pour 89.95 $ la liste de tous les appels du général Wesley Clarke, ancien commandant suprême de l'OTAN en Europe et ancien candidat à l'investiture démocrate à la Maison Blanche en 2004, ainsi que des données techniques concernant ces appels et son compte téléphonique.

Il est assez facile d'imaginer comment ces données peuvent, de différentes manières, être utilisées par des terroristes ou une nation étrangère. Il ne s'agit pas là d'un fait anecdotique, mais bien d'une tendance forte dans le domaine de la téléphonie mobile, en particulier aux États-Unis, où la loi permet légalement d'acheter ces données. Les sociétés de téléphonies, de services Internet, de grands groupes… pratiquent également l'externalisation de services critiques dans des pays à faible coût de main d'œuvre et peuvent ainsi être facilement victimes de telles collectes frauduleuses de données.

AMERICAblog just bought General Wesley Clark's cell phone records for $89.95

by John Aravosis (DC) - 1/12/2006 01:57:00 PM ET

I reported the other day that your cell phone records are on sale online for anyone to buy, without your permission. Well, this morning AMERICAblog bought former presidential candidate, and former Supreme Allied Commander of NATO (SACEUR), General Wesley Clark's cell phone records for one hundred calls made over three days in November 2005, no questions asked. (Clark's cell phone provider is Omnipoint Communications, which seems to be related to T-Mobile.)

All we needed was General Clark's cell phone number and our credit card, and 24 hours later we had one hundred calls the general made on his cell phone in November. The calls included a number of calls to Arkansas, to foreign countries, and at least one call to a prominent reporter at the Washington Post. To ensure that we actually had General Clark's correct cell phone number, we called the number this morning and the voice mail recording that answered said:

"Hi, this is Wes Clark, leave a message [unintelligible]."

We have subsequently called that number and spoken to a real person to confirm its authenticity, and to make sure General Clark was aware of this issue and what we were doing.

3

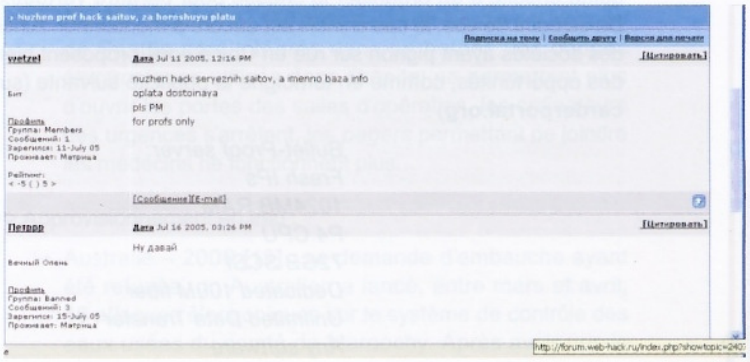
- Sur le marché de Gorbushka, à Moscou, il est possible de se procurer, pour une somme modique des DVD contenant des informations également sensibles, de nature à permettre ensuite de faire pression sur les intéressés par exemple :
  - La liste et les informations concernant les transferts en liquide de la Banque Centrale de Russie, pour 1500 $;
  - Les relevés d'imposition et les données personnelles afférentes (revenus, adresses…) pour 215 $;
  - Liste des clients de sociétés de téléphonie mobile pour 43 $;
  - État-civil complet, numéro de passeport, adresses, numéros de téléphone, description des véhicules… de tous les conducteurs de Moscou pour 100 $.
- Aux États-Unis, se procurer des informations personnelles permettant de se créer une identité fictive est très facile. Selon le groupe *Privacy Rights Clearinghouse*, plus de 89 millions de ces données ont été dérobées entre février 2005 et juin 2006. Ces données sont ensuite négociées sur certains sites et en particulier les numéros de sécurité sociale (SSN) et les informations attachées : 35 à 45 $ (sur les sites **secret-info.com** ou **iinfosearch.com**). Il faut savoir qu'aux États-Unis près de 50 % des établissements financiers n'utilisent que le SSN aux fins d'identification (source Unisys). On imagine facilement l'usage qui peut en être fait par des personnes peu scrupuleuses.

Ces quelques exemples, parmi de nombreux autres, illustrent suffisamment le propos pour comprendre que l'offre est riche dans ce domaine. Ajoutons que le vol ou la perte de clefs USB, de téléphones ou de PDA, d'ordinateurs portables… est également une source extrêmement riche d'informations sensibles, susceptibles d'être négociées. Encore trop de ces supports ne sont pas dotés de moyens de chiffrement, ce qui se révèle dramatique en cas de perte ou de vol. N'oublions pas les sociétés ou administrations qui revendent leurs disques durs dans l'espoir d'en tirer quelques euros, et ce, sans les formater de manière sécurisée. Un simple achat d'un disque dur usagé (sur le net, aux puces…) peut se révéler une véritable mine d'or.

1.2 Les cyber-mercenaires

Comme pour leurs homologues conventionnels, il est très difficile d'avoir des données précises sur leur nombre et leur réelle activité. Ce qui est sûr est qu'ils existent bel et bien. En revanche, leur niveau d'organisation, qui les emploie, à quelles fins… toutes ces données sont difficiles à obtenir, en supposant qu'elles le soient en ce qui concerne le domaine du numérique. On a souvent évoqué lors du conflit russo-estonien, voire des affaires d'espionnage attribuées à la Chine cet été [18], l'implication de groupes de pirates rémunérés ou sympathisants dans la réalisation de ces attaques.

Ainsi, le 15 juillet 2005 un post signé « Wetzel » cherchait une personne capable de pénétrer des sites professionnels, à des fins de collecte de renseignement de ses bases de données. Il offrait un paiement « adéquat » pour ce service en précisant que seuls des « professionnels » étaient souhaités. Une réponse parvint dès le lendemain (voir Figure 4) et, cinq jours plus tard, une autre demanda de plus amples précisions sur l'offre.



4 *Recherche de « cyber-mercenaires » sur un forum*

Cet exemple, parmi de nombreux autres, illustre la tendance, notamment sur des sites comme **Web-hack.ru** et des sites similaires. Il montre également que le marché souterrain propose plus que de simples numéros de cartes bancaires.

Le cas certainement le mieux connu est très probablement celui de la « location » de botnets pour réaliser des attaques et en particulier des dénis de service, des infections de cibles ou de la collecte de renseignements. Ainsi, il est possible de louer tout ou partie d'un botnet. Un réseau malicieux de 10 000 machines zombies se négocie actuellement environ 1000 $ l'heure. Un nombre impressionnant de services sont disponibles à qui peut les payer. Parmi eux (la liste n'est pas exhaustive) :

- désactivation de pare-.feu et/ou antivirus (dans le cas d'attaques en plusieurs vagues par exemple), blocage de sites d'éditeurs d'antivirus, modification des bases de mise à jour ;
- collecte de données soit par analyse de trames, soit après pénétration de systèmes ;
- vol de mots de passe (par exemple Spybot passe par la fonctionnalité `WNetEnumCachedPasswords`) ;
- DoS et DdoS ;
- Interception et redirection de trafic…

Selon Les Hynds (2005), directeur de la UK National Hi-Tech Crime Unit, « le marché de la location de botnets est devenu une industrie à part entière, contrôlée par le crime organisé ». Ces propos ont été confirmés lors de la conférence *Virus Bulletin* (session de clôture) en octobre 2007 par les responsables de cette unité, le FBI et l'OCLCTIC australien.

Selon ces responsables, l'activité des cyber-mercenaires est clairement prouvée, mais lorsque des questions plus précises ont été posées lors de cette conférence, ces responsables évoquent la sécurité nationale de leurs pays respectifs.

Il est néanmoins possible de se faire une idée relativement cohérente du recours à ces cyber-mercenaires. Dans un domaine comme celui de l'informatique, où certes d'un côté il est possible d'effacer la plupart de ses traces, le principe de précaution doit également partir du principe que certaines de ses traces ne sont pas accessibles dès lors qu'elles sont sur des serveurs étrangers, hors d'atteinte. Le recours à des « soldats de fortune » numériques – ou leur manipulation –, dont les motivations vont de l'appât de gain le plus vil à l'idéal le plus noble, représente donc un écran pratique derrière lequel un état voyou, par exemple, peut se dissimuler, quitte ensuite à en organiser la chasse pour paraître vertueux. De ce point de vue, le cas chinois est encore à mentionner. Ainsi, des sociétés ayant pignon sur rue en Chine vous proposent bien des opportunités, comme en témoigne la publicité suivante (sur **carderportal.org**) :

Bullet-Proof server:
Fresh IPs
1024MB RAM
P4 CPU
72GB SCSI
Dedicated 100M fiber
Unlimited Data Transfer
Any software
Based China
US$599.00 monthly
May use the server for:
Bulk web Host
Direct Mailing
We also supply e-mail list according to your order and sending out your message for you.
Hope to service for you.

Ainsi pour 20 $ par mois, il est possible d'héberger votre serveur de spam ou pour 30 $ par mois vos activités frauduleuses, et ce, avec les meilleures protections possibles, en premier la cécité des autorités chinoises. On imagine alors facilement l'intérêt de ce type de « service » pour un pays qui voudrait monter une attaque en prenant la Chine comme écran… ou la Chine voulant attaquer un pays profitant de ce genre de faux-semblants. Il est facile de comprendre pourquoi il n'est pas véritablement possible de savoir si une attaque venant de Chine est vraiment chinoise.

2. Protéger les infrastructures contre le risque informatique ?

Le marché du *malware* sous toutes ses formes est florissant, on vient de le voir. Cela offre d'autant plus de possibilités à toute personne désirant mener une attaque informatique, pour peu qu'elle dispose des fonds nécessaires (encore que, dans certains cas, les prix soient dérisoires).

Les attentats du 11 septembre, puis ceux de Madrid (2004) et Londres (2005) ont redonné une priorité à la protection des infrastructures dites « critiques ». Si leur protection physique est généralement bien prise en compte dans les plans gouvernementaux, qu'en est-il des risques liés à la menace informatique qui peuvent peser sur elles ?

2.1 Les Infrastructures critiques

« Infrastructure critique » est une expression générique utilisée pour décrire des biens physiques essentiels au fonctionnement d'une société et de son économie. Les différents pays ont une définition qui reprend généralement ces termes. Cela englobe donc une multitude de secteurs qui recouvrent généralement les domaines suivants :

- production et distribution d'énergie (électricité, gaz naturel, fuel…) ;
- télécommunications, technologies de l'information ;
- approvisionnement en eau ;
- agriculture, production et distribution alimentaire ;
- transport (ferroviaire, aérien, routier, approvisionnement en carburant…) ;
- services financiers ;
- services publics (police, armée, services d'urgence…) ;
- …

L'informatisation croissante et l'évolution de la société vers une société de l'information a maintenant conduit à étendre le terme non seulement aux infrastructures elles-mêmes, mais aussi à certains services qu'elles peuvent offrir. Les infrastructures informatiques sont maintenant souvent considérées comme critiques.

Les États-Unis, en particulier, le précisent explicitement, que ce soit dans le *Patriot Act* ou le *National Infrastructure Protection Program* (NIPP).

2.2 Le risque informatique contre les infrastructures critiques, mythe ou réalité ?

Bien évidemment, la protection de telles infrastructures est largement prise en compte par les pouvoirs publics dans le cadre de plans de protection contre les désastres naturels, le risque terroriste…

La menace informatique contre ces infrastructures n'est cependant pas à exclure. Les opinions quant à leur réalité ou leur efficacité potentielle divergent : pour certains (généralement dépendant d'organismes officiels), c'est une réalité dont les conséquences pourraient être graves, voire catastrophiques. Pour d'autres, elles sont très improbables et/ou relèvent du fantasme [4] et n'ont aucune chance de se produire et d'aboutir à un résultat tangible.

Le risque semble toutefois bien réel.

2.2.1 Des alertes nombreuses

Si de telles attaques intentionnelles sont rarement avérées, les exemples ne manquent pourtant pas quant à la possibilité quelles surviennent un jour :

- Infrastructure de transport :
  - Aéroport du Massachussetts, 1997 [5] : une effraction informatique met hors service le système de communication de la tour de contrôle pendant plusieurs heures. 60 000 foyers proches de l'aéroport sont également privés de téléphone.
  - Aéroport international d'Anchorage, 2006 [6] : un site Internet islamiste met en ligne ce qui **semble** être des images prises en direct par les caméras de l'aéroport (Figure 5) et propose un lien vers un programme d'administration distante **supposé** permettre d'en prendre le contrôle.

5 *Le 17 octobre 2006, un site islamiste a mis en ligne un message intitulé « Vous pouvez espionner les aéroports ennemis en contrôlant directement l'orientation des caméras ». Le site met un lien vers ce qu'il prétend être des vues en direct de l'aéroport international d'Anchorage (http://209.193.48.89/view/index.shtml). Le message donne des indications sur la manière de contrôler les caméras et promet de mettre en ligne des liens vers d'autres aéroports.*

Source : Middle East Media Research Institute. (http://memri.org/bin/articles.cgi?Page=archives&Area=sd&ID=SP132606).

La réalité de la chose n'est (et ne sera sans doute jamais avérée). Ce genre de lien ne reste jamais actif très longtemps. Le doute subsistera toujours… mais, jusqu'à présent, ce genre d'action « psychologique » parait être le mode d'action préférentiel de ces « groupuscules ».

- Réseau ferré géré par CSX (USA) – août 2003 [7] : le ver informatique Sobig met hors service le système informatique de la société CSX affectant la circulation des trains dans 23 états à l'est du Mississipi. Des retards allant jusqu'à 6 heures ont été occasionnés, une vingtaine de trains ont été annulés dans la région de Washington.

- Énergie :
  - Centrale nucléaire de Davis-Besse (Ohio – USA) – août 2003 [8] : le ver Slammer réussit à s'introduire sur le réseau de la centrale, paralysant des systèmes de contrôle important pendant près de 5 heures.
  - GAZPROM, 1999 [9] : bénéficiant de complicités internes, des pirates ont réussi à infiltrer le réseau informatique de la société et à prendre le contrôle, à l'aide d'un cheval de Troie, du système de régulation des flux de gaz naturel dans les pipe-lines.
  - CHEVRON, 1992 [10] : un ancien employé de la société CHEVRON réalise une effraction dans le système informatique et le reconfigure de manière à ce qu'il ne transmette plus les alertes. Cela a affecté les sites chimiques de CHEVRON dans 22 états et au Canada. Le sabotage sera découvert lorsque, lors d'un incident chimique mineur à la raffinerie de Richmond, les alertes ne seront pas transmises.

- Santé publique :
  - Seattle – 2005 [11] : 150 ordinateurs du *Northwest Hospital* de Seattle sont intégrés dans un botnet : les badges électroniques utilisés dans l'hôpital ne permettent plus d'ouvrir les portes des salles d'opération, les ordinateurs des urgences s'arrêtent, les *pagers* permettant de joindre les médecins ne fonctionnent plus…

- Approvisionnement en eau :
  - Australie – 2000 [12] : sa demande d'embauche ayant été refusée, un Australien a lancé, entre mars et avril, 46 attaques électroniques sur le système de contrôle des eaux usées du comté de Maroochy. Après avoir pris le contrôle du système, il a déversé quelques millions de litres d'eaux usées, polluant rivières et parcs de la région.
  - États-Unis – 2002-2006 [13] : un pirate informatique a pénétré en février 2006 dans le système de contrôle d'une station d'épuration près de Harrisburg et y a implanté des logiciels malicieux. Le pirate est passé par l'ordinateur portable d'un employé et a utilisé ensuite le compte d'accès distant de ce dernier pour s'introduire dans le système informatique de la station. Il s'agissait là, aux États-Unis, du quatrième incident visant ces infrastructures en quatre ans : attaque en déni de service, accès et prise de contrôle du système de commande du traitement des eaux en Californie, affichage de messages contre la guerre en Irak.

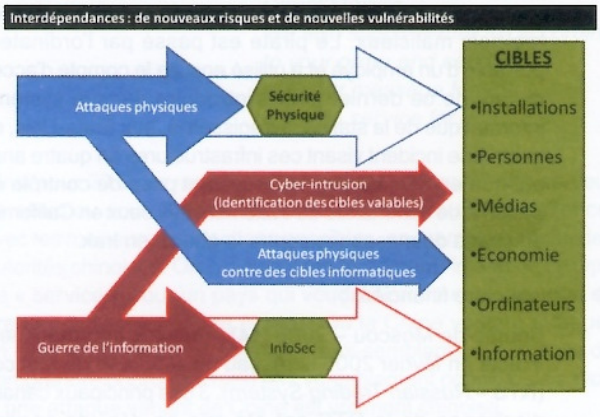
- Infrastructure financière :
  - Bourse de Moscou – 2006 [14] : un virus informatique a infecté en février 2006 le réseau de la bourse de Moscou (RTS – *Russian Trading System*). 3 des principaux canaux d'échange de la RTS ont été mis en déni de service obligeant à une interruption des cotations pendant une heure. La possibilité d'une attaque ciblée, dont le but n'est toujours pas très clair, a été évoquée.
  - Réseaux bancaires : les exemples ne manquent pas d'attaques (généralement à des fins d'escroquerie) contre des banques (réseaux informatiques, distributeurs de billets…)

- Services de sécurité et administration : les exemples de pénétration des réseaux informatiques militaires aux États-Unis ne manquent pas (Pentagone (juin 2007), déconnexion de 1500 comptes de messagerie – *Naval War College* (2006), aucune connexion à l'Internet pendant plusieurs semaines – vol du logiciel de planification des vols des hélicoptères de l'armée de terre et de l'armée de l'air…). Sans oublier l'Estonie, dont il a déjà été largement question.
- Information : il n'est pas besoin de rappeler les attaques qui ont visé les serveurs DNS en 2005 et 2007 ou l'effet de saturation de Slammer qui, en janvier 2003, a fortement ralenti Internet (privant même la Corée du Sud d'accès pendant 24 heures) pour se rendre compte que les infrastructures de l'information sont également attaquées.

Il apparaît clair que de telles attaques restent possibles. Elles n'ont été jusqu'à présent, dans la majorité des cas, que le fait du hasard et ne correspondaient pas à une volonté avérée de s'attaquer à ces infrastructures.

2.3 Une interconnexion croissante

La majeure partie de ces infrastructures dépend de systèmes de contrôles informatiques (SCADA - *Supervisory Control And Data Acquisition*). Les possibilités de gestion et de supervision à distance permettent en effet de centraliser et de réduire les coûts globaux de gestion. La contrepartie est une vulnérabilité accrue des systèmes liée aux risques informatiques : gestion des vulnérabilités logicielles, toujours présentes, piratage, attaques en déni de service, cette supervision se faisant par l'Internet. Cette vulnérabilité est accrue par l'imbrication de l'informatique en tant que cible ou vecteur d'attaque (Figure 6). Si le risque pesant sur ces systèmes a longtemps été rattaché à une menace interne, la multiplication de ces systèmes de gestion, l'interconnexion croissante, la possibilité d'accès distant offerte de plus en plus souvent aux employés ont fait basculer l'origine des attaques qui peuvent maintenant venir de l'extérieur.

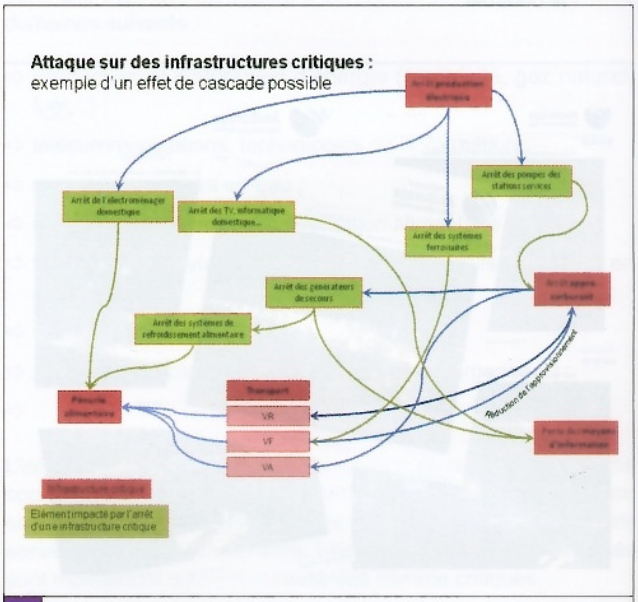


6 *(Source : PCCIP : Critical foundations – Protecting America's infrastructures – 1997) – En 1997 déjà, le rapport de la President's Commission on Critical Infrastructure Protection (PCCIP) pointait du doigt l'importance croissante de l'informatique dans la protection des infrastructures critiques, d'une part en termes de dépendance, d'autre part en tant que moyen de renseignement ou en tant qu'arme. Cette étude soulignait que la protection des infrastructures critiques devrait être examinée sous l'angle de la sécurité nationale compte tenu des conséquences que pourrait avoir une attaque.*

2.4 Tous ces systèmes sont attaquables

L'évaluation du risque et de la menace contre ces infrastructures est à faire sérieusement et objectivement. S'il ne sert à rien de crier « au loup », il n'en faut pas moins envisager toutes les possibilités. Le risque physique, s'il semble le plus sérieux, ne doit pas faire oublier que de nombreux systèmes comptent (pour ne pas dire reposent) de plus en plus sur les moyens informatiques pour leur gestion et leur sécurité.

L'analyse des risques et des conséquences d'une défaillance (accidentelle ou malveillante) est essentielle et doit se faire de manière globale de façon à tenir compte de l'interdépendance entre ces différentes infrastructures. L'effet de cascade reliant ces différents éléments n'est pas toujours facile à évaluer : l'attaque sur une infrastructure critique peut entraîner des répercussions sur d'autres et amplifier ainsi les dégâts commis (Figure 7).



7 *Cet exemple illustre (de façon simplifiée) ce que pourrait être l'effet de cascade lié à un arrêt prolongé de la production électrique : désagréments liés à l'absence d'électricité (lumière…), plus de stations service (pompes arrêtées). L'absence d'électricité provoque également l'arrêt du trafic ferroviaire (plus de signalisation, les aiguillages ne fonctionnent plus…). Il en résulte progressivement une pénurie alimentaire, une perte des moyens de communication… D'autres conséquences ne sont pas évoquées sur ce schéma volontairement simpliste (impact sur les institutions financières par exemple : arrêt des réseaux bancaires, arrêt des distributeurs de billets…). Des expériences récentes aux États-Unis ont mis en évidence la vulnérabilité de l'infrastructure électrique nationale à une attaque d'origine informatique (voir encadré 2)*

La dépendance extrême vis-à-vis d'Internet est telle qu'une « simple » paralysie de ce réseau, même de manière limitée dans le temps, aurait des conséquences économiques telles, qu'à côté, la crise de 1929 ne ressemblerait qu'à une simple promenade de santé. Le problème pour les experts n'est plus de savoir si cela est possible, mais plutôt quand cela arrivera. Des chercheurs canadiens [15] ayant étudié le protocole BGP (*Border Gateway Protocol*) y ont identifié un grand nombre de faiblesses qu'il serait facile d'exploiter dans ce but (dont certaines déjà exposées par N. Dubée [21]).

encadré 2

Le risque informatique contre l'infrastructure électrique aux États-Unis. [16]

En mars de cette année, des chercheurs américains, étudiant les risques liés à une vulnérabilité découverte dans un système de supervision des générateurs utilisés dans la majeure partie des centrales électriques américaines, se sont rendu compte qu'il était possible d'amener un générateur à s'autodétruire en piratant son système de contrôle, celui-ci renvoyant des informations correctes à l'opérateur et des instructions erronées au générateur.

Certes, cette attaque a été réalisée dans des conditions expérimentales qui l'ont facilité. Il n'en demeure pas qu'elle a montré, de manière peut-être plus visible que jamais auparavant, les conséquences physiques que peut avoir une attaque informatique. Ce test à rappelé, aux États-Unis, les conclusions auxquelles étaient arrivés des conseillers du président Bush quelques années auparavant : une organisation telle qu'un service de renseignement étranger ou un groupe en relation avec la mouvance terroriste « pourrait mener une attaque électronique structurée contre l'infrastructure électrique [des États-Unis], avec un haut degré d'anonymisation et sans mettre un pied dans le pays ».

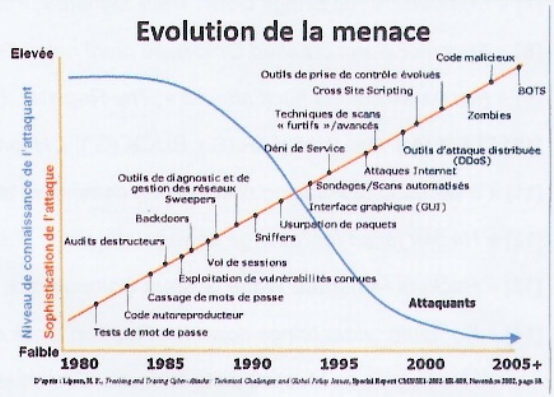
Depuis, la vulnérabilité utilisée a été corrigée, mais si Robert Jamison, sous-secrétaire d'état au *Department of Homeland Security* (DHS), indique que le risque a été grandement réduit, il ajoute que ce genre de vulnérabilité ne peut pas être totalement éliminé. Le problème, dans ce cas précis, est que ces systèmes sont de plus en plus connectés à l'Internet pour des raisons d'efficacité, ce qui les rend plus vulnérables encore.

De nombreux experts, tant du monde du renseignement, que de l'industrie, de l'économie, de la sécurité se sont exprimés sur le sujet. Joseph Weiss, expert en sécurité, a souligné devant le Congrès, que « de nombreux systèmes utilisés sont anciens. Si l'on parvient à y accéder, on peut leur faire faire ce que l'on veut ». Il a également souligné (interview accordée à CNN) que le risque était amplifié par le fait que ces mêmes systèmes (générateurs et système de gestion) ont été vendus à l'étranger à des nations qui connaissent sans doute la vulnérabilité employée et savent l'utiliser. « Ils ont le même entraînement [que nous], les mêmes mots de passe… ».

Au final, la question posée par cette expérience et les discussions qu'elle a soulevées est : « les États-Unis arriveront-ils à boucher toutes les vulnérabilités avant que les hackers ne les découvrent ? »

Il est généralement admis que la protection contre certains risques relève de la responsabilité de l'État (menace terroriste par exemple). Beaucoup d'infrastructures critiques relèvent du domaine privé (des estimations indiquent que 85% des infrastructures critiques des États-Unis dépendent du secteur privé ; ce taux avoisinerait les 90% en Allemagne). Les canaux utilisés en cas d'attaque informatique dépendent généralement de l'Internet (les fournisseurs d'accès acheminant le trafic). Il est dès lors logique qu'une gestion efficace du risque informatique passe par une concertation de tous les interlocuteurs impliqués à quelque degré que ce soit dans cet environnement : l'État, généralement responsable de la coordination d'ensemble, le propriétaire des infrastructures (secteur public ou privé, selon les cas) est généralement responsable de la mise en œuvre des moyens de protection. Il semble pertinent de considérer qu'une implication des fournisseurs d'accès Internet soit à envisager dans le cas du risque informatique : ils sont les possesseurs des canaux utilisés pour mener les attaques et peuvent avoir leur rôle à jouer dans le cadre d'attaques massives comme celles que l'on a pu voir dans le cas de l'Estonie

La plupart des pays ont une telle approche, avec un degré interventionniste de l'État plus ou moins important dans la supervision et la mise en place de mesures de protection complétant celles du « propriétaire ».



8 *D'après : Lipson H. F., Tracking and tracing cyber-attacks: technical challenges and global policy issues, Special report CMU/SEI-2002-SR-009, novembre 2002, p. 10.*
La mise à disposition d'outils performants et d'« offres de service » sur Internet permettent maintenant de mener des attaques sophistiquées sans pour autant avoir des connaissances importantes.
Si un bon niveau initial de connaissance technique est nécessaire, ce niveau de connaissance décroît rapidement avec le temps, au fur et à mesure que les outils sont mis sur le marché.
Parallèlement, et paradoxalement, les attaques utilisées sont de plus en plus complexes, même si elles peuvent être utilisées sans pour autant posséder de grosses connaissances techniques.

Conclusion

La méthode employée lors des attentats du 11 septembre était envisageable. Qui aurait pensé qu'elle serait utilisée un jour ? Cette simple question n'est pas sans conséquence : des attaques informatiques sur les infrastructures critiques sont possibles, de nombreux « accidents » nous l'ont montré. Peut-on prendre le risque, qu'un jour, quelqu'un cherche intentionnellement à les attaquer sans s'être préparé à les défendre ? Certes, les exemples récents de tentative de « cyber-terrorisme » n'ont pas montré une capacité de destruction réelle, ni une efficacité particulière et les logiciels d'attaque largement diffusés à ces occasions (Estonie, « Cyber-Jihad ») relèvent parfois plus du script-kiddy que de moyens informatiques évolués et efficaces (vente de virus performant, offre de service, botnets...). Rien ne dit qu'une telle attaque aura lieu un jour. Rien ne garantit non plus qu'elle n'ait jamais lieu. Rien ne peut également assurer que les méthodes et moyens mis en place seront efficaces s'il s'en produit une. Doit-on pour autant ne pas considérer ce risque ? Si « se faire battre est excusable, se faire surprendre est impardonnable » [17]. Le principal problème étant que l'on ne sait pas si l'adversaire dispose d'armes efficaces dans ce domaine tant qu'il n'a pas commencé à les utiliser.

Notes

[1] Bien que nous ne soyons pas adeptes de la mention « source anonyme bien informée », notre correspondant a cependant accepté de nous communiquer certains éléments sous couvert d'anonymat. Proche des milieux techniques officiels du gouvernement américain, ses informations ont toutefois été partiellement recoupées par nous (sur des sujets autres que l'Estonie) à l'aide de différentes autres données techniques plus ou moins publiques et éparses, malheureusement trop nombreuses pour être citées ici. Nous sommes dans l'attente de documents complémentaires sur l'Estonie que celui-ci doit nous transmettre.

[2] On appelle « conflit de basse intensité » ce qui autrefois était appelé « guérilla » ou « petite guerre ». Il se distingue par son faible niveau de violence conflictuelle, n'obéissant pas nécessairement à des schémas conventionnels, avec un engagement de forces limité, sans ligne de front clairement définie, comportant des zones de combat réduites, mais éventuellement multiples, de nature asymétrique (rapport du faible au fort) et pouvant faire intervenir des forces de nature non clairement identifiées comme militaires (au regard du droit des conflits armés). Merci au LCL Saboya (Thierry, pas Yann) pour cette définition éclairée !

[3] Il y a en Chine 30 000 à 50 000 groupes de police dédiés à la surveillance d'Internet, répartis dans 700 villes, avec comme priorité... la surveillance et la traque des sites prônant la démocratie en Chine !

[4] « *Myths of cyberwar – Information Security* », avril 2004. – « *Electronic Pearl Harbor for a Day* », **dailyscare.com**

[5] « *Teen hacker faces federal charges* », **CNN.com**, 18 mars 1998.

[6] « *Mujahideen Gather Information on Anchorage International Airport* », MEMRI, *Islamic Websites Monitor* n°9/*Special Dispatch Series*, No. 1326, 18 octobre 2006

[7] « *Computer Virus Brings Down Train Signals* », *InformationWeek*, 20 août 2003.

[8] « *Slammer worm crashed Ohio nuke plant net* », *The Register*, 20 août 2003.

[9] « *Russia welcomes hack attacks* », *The Register*, 27 avril 2000.

[10] CLEM (A.), GALWANKAR (S.), BUCK (G.), « *Health implications of cyber-terrorism* ».

[11] « *3 accused of inducing ill effects on computers at local hospital* », *The Seattle Times*, 11 février 2006.

[12] « *Hacker jailed for revenge sewage attacks* », *The Register*, 31 octobre 2001.

[13] « *Hackers Penetrate Water System Computers* », *ABCNews*, 30 octobre 2006.

[14] « *Computer virus brings down the Russian stock exchange* », *TechShout.com*, 05 février 2006. –

« Virus : la Bourse de Moscou a été attaquée. Escroquerie ? », **Silicon.fr**, 08 février 2006.

[15] Evangelos Kranakis et Bill Aeillo du *School of Computer Science*, Carleton University, Ottawa.

[16] « *Staged cyber attack reveals vulnerability in power grid* », **CNN.com**, 26 septembre 2007 et le reportage vidéo associé.

Control Systems Cyber Security – The need for appropriate regulations to assure the cyber security of the electric grid – Joseph Weiss, (*testimony before the Committee on Homeland Security's subcommittee on emerging threats, cybersecurity, and science and technology*), 17 octobre 2007.

[17] Citation attribuée à Napoléon.

Références

[18] EVRARD (P.) et FILIOL (E.), « Guerre, guérilla et terrorisme informatique : fiction ou réalité ? », *MISC – Le journal de la sécurité informatique* 33, pp. 09-17, septembre 2007.

[19] KOSTYA (Kortchinsky), « 0-Day », Actes de la conférence *SSTIC* 2005.

[20] LACOMBE (E.), RAYNAL (F.) et NICOMETTE (V.), « De l'invisibilité des rootkits : application sous Linux », Actes de la conférence *SSTIC* 2007, pp. 145-178.

[21] DUBÉE (N.), « BGP et DNS : attaques sur les protocoles critiques de l'Internet », Actes de la conférence *SSTIC* 2003.

[22] LEYDEN (J.) « *Cybercrime 'more' lucrative than drugs* », *The Register*, 29 novembre 2005, **http://www.theregister.co.uk/2005/11/29/cybercrime/**

1/5

Autopsie informatique : les 4 approches à connaître

Recouvrement, e-discovery, incidents, forensique : les méthodes d'analyse

Fonctionnellement et sémantiquement, on divise généralement « l'analyse d'ordinateur » en quatre activités distinctes :

- **la récupération de données (aussi appelée « recouvrement de données ») ;**
- **l'e-discovery ;**
- **l'intervention sur incident (incident response) ;**
- **l'analyse forensique.**

Ces quatre activités utilisent souvent des outils et méthodes similaires, et c'est précisément pour cette raison qu'il convient de bien les définir pour en comprendre les différences sans faire de confusion.

mots clés : *recouvrement / récupération / forensique / e-discovery*

1. La récupération de données

La récupération de données fait généralement suite à un accident fortuit et involontaire.

Cet accident peut être purement matériel ; le cas le plus fréquent est une panne de disque dur, éventuellement dans un contexte de sinistre plus global (incendie, dégât des eaux, effondrement suite à un attentat, etc.).

L'accident peut également être logiciel, et, dans ce cas, les origines sont extrêmement variées : effacement involontaire de fichiers, formatage intempestif, écrasement partiel de données tierces par l'enregistrement en continu d'un « gros » fichier (vidéo, copie bit à bit, etc.), corruption d'un fichier qui « explose » à force de grossir (fichier Exchange de plusieurs téra-octets, fichier Powerpoint contenant plusieurs milliers d'images, base de données, fichier qui dépasse la taille maximale autorisée par les spécifications du système de fichiers, etc.), corruption de la structure du support numérique (effacement du *master boot record* par un virus, effacement du secteur de boot sur une clé USB, etc.)

Un malheur n'arrivant jamais seul, ces accidents s'accompagnent évidemment d'une absence de sauvegarde ou d'une corruption de la sauvegarde elle-même !

En cas d'appel à un tiers (société spécialisée dans le recouvrement de données), le coût du service réserve généralement de facto ce genre de prestation aux données critiques des professionnels (base de données clients, brevets et innovations, comptabilité, œuvre littéraire ou musicale en cours d'écriture, etc.), bien que certains particuliers y aient également parfois recours (notamment pour la récupération de travaux de thèse, de photos de mariage, de voyage de noces, de baptême, etc.). Les risques de fuite de données n'étant pas nuls, le Secrétariat Général pour la Défense Nationale (SGDN) a promu, dans la cadre de sa mission de lutte contre les menaces liées à l'intelligence économique, une charte de bonnes pratiques signée à ce jour par six entreprises œuvrant sur le territoire national [1].

Dans les cas les plus élémentaires (effacement ou formatage accidentel provoquant la perte de fichiers « classiques » de type bureautique, image, vidéo, etc.), l'emploi de logiciels publics simples d'utilisation, gratuits ou payants [2] peut s'avérer suffisant et relativement efficace.

Dans les cas plus complexes, il convient de procéder à une réparation physique du disque dur (ou plus généralement du support de données [3]), à une reconstruction du master boot record, du secteur de boot ou de la table d'allocation des fichiers, ou encore à un réassemblage des fragments du fichier perdu ou à une remise en forme du contenu brut du fichier corrompu (sous réserve, par exemple, que la structure interne dudit fichier soit connue).

La réparation physique du disque dur peut consister en un « simple » remplacement de pièces mécaniques ou électroniques défectueuses (carte-contrôleur, tête de lecture-écriture, moteur d'entraînement des plateaux), moyennant la disponibilité d'une collection de disques durs de rechange et des conditions ambiantes adaptées (salle blanche – ou plus simplement hotte à flux laminaire, dispositifs antistatiques de dissipation de l'électricité statique).

La réparation physique peut aussi nécessiter le nettoyage d'éventuelles salissures ou polluants divers, tels que poussières, suie, eau (douce ou de mer), boue, rouille superficielle, etc. Les sprays pour cartes électroniques souillées, une solution d'alcool liquide très volatile (astuce bien connue des plongeurs), une étuve (avec température réglable !) et un bac à ultrasons s'avèrent alors utiles.

Mais la réparation « physique » peut souvent se traduire également par une opération plus ardue de réparation de modules corrompus du *firmware* embarqué, les symptômes les plus fréquents étant un « claquement » répété du disque dur, une non-reconnaissance du disque par le système et/ou l'annonce par le disque de caractéristiques incohérentes (ex. : capacité de stockage aberrante). Rappelons que le firmware sert (entre autres) à gérer les fonctionnalités « bas-niveau » du disque dur, telles que le SMART [4], le HPA [5] et le DCO [6], le Security Mode (mot de passe), etc., et que ce firmware est, contrairement à une idée fort répandue, principalement stocké sur les plateaux eux-mêmes, dans des zones de service (*service area*) en adressage négatif [7], la carte-contrôleur ne jouant là qu'un rôle assez secondaire. Un exemple typique de corruption de firmware est le débordement de la G-List [8] qui vient écraser des « modules vitaux ». Bien entendu, des outils professionnels spécialisés [9] sont indispensables, les commandes d'accès à ces zones de service étant généralement propriétaires et non documentées ; il est également essentiel de disposer d'une base de firmwares « de rechange ».

Le principe essentiel du recouvrement de données est (en caricaturant à peine) l'absence de prise de connaissance du contenu des fichiers : soit on extrait en vrac et sans tri les données

Chef d'escadron **Nicolas DUVINAGE**
Chef du département informatique-électronique (INL)
Institut de recherche criminelle de la gendarmerie nationale (IRCGN)

du disque dur ou du système de fichiers réparé (car le client souhaite récupérer intégralement le support numérique – ou au minimum une copie parfaite de celui-ci – dans son état « pré-accident »), soit on récupère de façon très ciblée des données bien identifiées et bien connues du client (tous les fichiers Word, tous les fichiers créés ou modifiés entre la date t_1 et la date t_2, le fichier portant tel nom, etc.). Dans tous les cas, on sait parfaitement et avec précision ce que l'on recherche, et on a la certitude de l'existence des données sur le support numérique avant l'accident.

Quant à la présence de contenus chiffrés sur le support numérique, elle ne pose a priori aucune difficulté, puisque le client est généralement à même de fournir les clés ad hoc en cas de besoin (sauf si la mission consiste précisément à récupérer le contenu d'un fichier chiffré dont la clé a été perdue **[10]**...).

La seule difficulté (mais elle peut être de taille !) est donc la réparation elle-même, qui peut durer de quelques minutes à plusieurs jours (en particulier pour la récupération de « gros » fichiers corrompus).

2. L'e-discovery

L'*e-discovery* est une discipline encore jeune, venue tout droit des États-Unis, avec laquelle le rédacteur est peu familier. En résumé (les puristes pardonneront ce raccourci), il s'agit d'un véritable « droit de perquisition privé », qui s'applique principalement entre entreprises en conflit : le plaignant mandate un expert, qui a alors un droit quasi illimité de fouille dans les systèmes d'information du défendeur, et qui a pour mission de rechercher tout fichier, email, base de données, etc., en lien avec la plainte et susceptible d'apporter un éclairage dans le règlement du conflit. Toutes les machines « adressables » par des employés de l'entreprise visitée peuvent être inspectées (stations de travail, postes-clients, serveurs, baies de stockage, etc.), tant en local (bâtiments où se trouve physiquement l'expert mandaté) qu'à distance à travers un réseau de type intranet, extranet, VPN, accès Internet, etc. (sites secondaires, *data center*, données hébergées à l'extérieur, etc.).

Pour bien comprendre ce concept, on peut en rapprocher la philosophie de celle (bien connue des Français !) de l'huissier de justice mandaté pour saisir des biens en vue du règlement d'une créance.

L'e-discovery n'existe en théorie qu'en droit américain (et dans une certaine mesure au Royaume-Uni), mais par extension toutes les entreprises américaines ayant des filiales (ou des données...) hors des États-Unis sont concernées, de même que les entreprises non américaines ayant des intérêts sur le sol des États-Unis.

La principale particularité technique de l'e-discovery consiste en la masse considérable de données à traiter en un temps limité (le temps de travail facturé par l'expert mandaté ne doit pas coûter plus cher que le préjudice subi, et le droit américain ne laisse pas un accès illimité dans le temps aux ressources de l'entreprise visée par la plainte) : ainsi n'est-il pas rare que des dizaines de téra-octets, voire des péta-octets **[11]** (!), doivent être triés en quelques heures, tout au plus en quelques jours, en espérant trouver l'unique email, LE compte-rendu de réunion interne ou LE projet de brevet éclairant les responsabilités des parties en conflit.

Il s'agit donc véritablement de chercher une aiguille dans une botte de foin : on comprend aisément qu'il n'est alors pas question de rechercher des données effacées ou de perdre un temps précieux à réparer un disque dur en panne alors que des centaines d'autres « bien vivants » attendent ! On est là dans le traitement de masse, pas dans la dentelle (sauf si les recherches ont pu se focaliser sur une machine en particulier) !

Par opposition à la récupération de données, on ne sait pas forcément exactement ce que l'on cherche, on cherche « tout ce qui peut être utile » (sauf si le plaignant recherche un document bien spécifique).

La méthode consiste ainsi, très schématiquement, à comprendre vite et bien l'architecture réseau de l'entreprise, à identifier les machines (stations ou serveurs) susceptibles de contenir les données les plus intéressantes, sans oublier les données stockées à distance (éventuellement externalisées), avant d'effectuer une première recherche grossière par critères tels que nom de fichier, type de fichier, dates de création ou modification de fichier, mots-clés, emplacement de fichier dans l'arborescence, etc. L'utilisation d'outils spécialisés dans la « fouille réseau **[12]** », capables en outre de visualiser le contenu de fichiers très hétérogènes, est conseillée.

3. L'intervention sur incident (incident response)

L'intervention sur incident **[13]** (*incident response*), que ce dernier soit réel ou fictif, est sans doute l'activité la plus familière aux lecteurs de MISC, et ne sera donc pas abordée en détail. Elle est notamment l'apanage des spécialistes et responsables SSI. Il s'agit grosso modo d'analyser une machine qui a été compromise, à l'insu de son propriétaire ou avec son plein assentiment (*honeypot*, audit, challenge, etc.), par un humain ou par un automate (ex. : ver), afin de comprendre l'enchaînement qui a conduit à la compromission : quelle(s) vulnérabilité(s) ou faiblesse(s) architecturale(s) du système a été exploitée, quel(s) outil(s) a(ont) été utilisé(s) pour exploiter la (les) faille(s), à quel moment, quelles ont été les conséquences potentielles ou réelles de la compromission (effacement de données, vol de données, interception de communications, déni de service, défaçage de site Web, etc.)...

Les objectifs de l'intervention sur incident sont principalement la cessation de la compromission et le rétablissement pérenne du fonctionnement normal du système, ce qui passe notamment par la « pose de rustines » (ou éventuellement la modification de fond de l'architecture du système). En tout état de cause et quel que soit le contexte de l'intervention sur incident, l'objectif général est de contribuer à améliorer la sécurité du système tout en assurant son bon fonctionnement.

L'identification de l'auteur de l'attaque n'est la plupart du temps pas prioritaire, sauf quand une riposte est envisagée, que celle-ci soit purement juridique (dépôt de plainte), ou purement technique (contre-attaque)... et dans ce cas purement illégale en droit français **[14]** !

Très souvent, l'intervention a lieu alors que le système attaqué est toujours en fonctionnement. Selon les cas et selon les besoins, la méthode consiste très schématiquement, sur la machine compromise et/ou sur les équipements réseau traversés (ex. : *firewall*), voire dans de très rares cas sur la machine attaquante, à lister, identifier et analyser les processus en cours de fonctionnement au moment de l'incident, les programmes qui se lancent automatiquement au démarrage ou de façon régulière et plus généralement toute application « suspecte », à lister les ports TCP/UDP ouverts et les applications qui les utilisent et à identifier les logs pertinents et à les analyser. Des outils de capture de la mémoire vive et de *parsing* du *dump* en résultant peuvent être utiles [15]. Lorsqu'un programme suspect a été trouvé, il peut être judicieux d'observer son comportement dans un environnement de test [16] (LAN isolé du reste du réseau, machine virtuelle, etc.), voire d'en faire la rétro-conception [17].

L'essentiel du travail est focalisé sur le système (d'exploitation et/ou d'information) lui-même, et assez peu sur les données-utilisateur qu'il contient (sauf pour vérifier qu'elles n'ont pas subi de dommages). Quand la compromission a résulté en une perte de données, on bascule dans l'activité « recouvrement de données ».

4. L'analyse forensique

L'analyse forensique, enfin, regroupe potentiellement toutes les activités mentionnées supra, mais dans un but et dans un contexte très différents.

Au sens de l'auteur, et plus généralement au sens de tous les acteurs des « forces répressives d'Etat [18] » (gendarmerie et police nationales, douanes, direction nationale des enquêtes fiscales, direction générale de la consommation, de la concurrence et de la répression des fraudes, etc.) et de la justice (experts judiciaires privés), regroupés au sein de diverses associations [19], l'analyse forensique est au service de la justice, pénale et parfois civile.

Plus généralement, les sciences forensiques (on utilise aussi le terme de « criminalistique [20] »), popularisées par des séries télévisées telles que *Les Experts*, consistent en l'étude scientifique des éléments de preuve saisis dans une enquête. Elles couvrent potentiellement l'intégralité du spectre scientifique, de la médecine légale (autopsies) à la biologie (ADN), en passant par la toxicologie (produits stupéfiants et psycho-actifs, poisons), la balistique, l'entomologie-palynologie (étude des insectes nécrophages et des pollens), l'étude des véhicules [21] (pneumatiques, peintures, etc.) et des faux documents, la comparaison de voix [22], etc. Bien entendu, l'informatique, l'électronique (analogique ou numérique), les réseaux et les télécommunications ne sont pas oubliés.

Contrairement à une idée fort répandue, les experts forensiques dans ces domaines *high tech* ne traitent pas seulement des éléments de preuve intervenant dans des enquêtes relatives à des « piratages informatiques ». On pourrait même dire qu'à quelques exceptions près [23], les atteintes aux systèmes de traitement automatisé de données (STAD [24]) ne représentent qu'un faible pourcentage de leur activité. En effet, le rôle des experts forensiques est plus généralement d'apporter une assistance technique dans toute procédure où sont utilisés (par exemple) un ordinateur ou un téléphone GSM, et où l'analyse de ces objets peut présenter un intérêt pour la manifestation de la vérité !

De tels objets sont aujourd'hui omniprésents dans tous les domaines de la vie : ainsi, le champ infractionnel couvre potentiellement l'intégralité du Code Pénal (pédopornographie, livres de comptes informatiques dans des cas de délinquance économique et financière, emails de menaces dans des cas d'homicides, réalisation de faux documents par ordinateur, communications par GSM ou *instant messengers* dans un réseau de trafic de stupéfiants, etc.), du *Code de la propriété intellectuelle* (contrefaçon de musiques, films, logiciels, livres, base de données telles que des annuaires, etc.), du *Code monétaire et financier* (ex. : contrefaçon de cartes bancaires), du *Code de la santé publique* (incitation à l'usage de stupéfiants sur Internet, publicité en ligne sur le tabac et l'alcool, vente en ligne de médicaments ou produits pharmaceutiques tels que Viagra, lentilles de contact, etc.), du *Code de la consommation* (conditions de ventes par correspondance/ventes en ligne, modification de l'IMEI des téléphones GSM, etc.), du *Code des douanes* (importation de cigarettes achetées sur Internet, contrefaçon de grandes marques, etc.), du *Code général des impôts*, etc.

Les experts forensiques peuvent intervenir quasiment à tout moment de la procédure. Leur soutien peut être nécessaire sur les lieux-mêmes de la scène d'infraction (scène de crime ou perquisition), pour aider à effectuer des constatations pertinentes et faire procéder à des saisies de matériels ou données dans des conditions techniques indiscutables. Ainsi, quand une perquisition est menée dans une grande entreprise, les méthodes et outils employés peuvent être très similaires à ceux utilisés en e-discovery, les contraintes de temps et de quantité de données à trier étant quasiment les mêmes. Mais les experts peuvent aussi, dans ce cas, utiliser des méthodes et outils d'intervention sur incident, en particulier lorsqu'ils rencontrent des machines en cours de fonctionnement (« machines vivantes [25] »), chez un particulier suspect ou en environnement professionnel : il peut en effet être utile de récupérer « en direct » dans la mémoire vive les *logins* et mots de passe d'accès à des serveurs distants en cours d'utilisation ; de même la présence de solutions de chiffrement intégral fort peut inciter l'expert à analyser le système en cours de fonctionnement, et non en « post mortem ».

Lors de l'audition (interrogatoire) d'un suspect ou d'un témoin, l'expert forensique peut être commis pour jouer le rôle d'un « traducteur technique », assistant les enquêteurs pour comprendre les termes et concepts employés (par exemple) par un responsable SSI « noyant le poisson », volontairement ou non. Son aide peut être décisive pour pointer dans une audition d'éventuelles incohérences qui auraient pu échapper à un enquêteur profane.

Plus rarement, l'expert peut être amené à participer à une reconstitution judiciaire (ex. : reconstitution d'un accident mortel dans une chaîne de production industrielle contrôlée par des automates informatiques).

La majeure partie du travail des experts forensiques consiste néanmoins en l'analyse *a posteriori* des éléments matériels ou immatériels (données [26]) saisis au cours de la procédure. Cette analyse ne se limite pas aux seuls disques durs d'ordinateurs. Non seulement elle concerne potentiellement tous les supports de stockage informatique, que leur technologie soit magnétique (bandes magnétiques de type DAT ou autres, disquettes, cartouches de type ZIP/JAZ/etc.), optique (CD et DVD de tous types), électronique (mémoire Flash : clés USB, cartes à mémoire de type SD/MMC/etc.) ou magnéto-optique (disques magnéto-optiques, etc.), mais aussi, plus généralement, tous les supports

numériques, courants et moins courants, susceptibles d'avoir été utilisés au cours d'une infraction et présentant un intérêt potentiel pour l'enquête : PDA, téléphones GSM et cartes SIM, lecteurs MP3, appareils photo numériques, dictaphones numériques, cadres photo, disques durs de photocopieurs, GPS, cartes bancaires, etc.

De façon générique, les missions types confiées à l'expert forensique sont :

- Rechercher des données, « actives », effacées, mais encore présentes dans le système de fichiers, ou présentes uniquement sous forme de résidus et de traces (dans le *slack*, dans les secteurs non alloués, etc.). Ces données peuvent être des données utilisateurs, issues d'une action volontaire et délibérée (ex. : contenu de fichiers de type email ou SMS, bureautique, multimédia, base de données, etc.) ou des données davantage « propagées automatiquement » à travers le système, à l'issue d'une installation ou d'une configuration (ex. : données permettant de concourir à l'identification d'un auteur, d'une victime, d'un propriétaire ou d'un utilisateur, telles qu'une adresse IP dans un en-tête d'email, un nom de machine ou de personne dans des métadonnées de fichiers, etc.).
- Retracer l'activité d'un utilisateur (historique d'utilisation de programmes spécifiques, historique des navigations sur Internet – quels que soient le protocole et le client utilisés, détails de l'activité éventuelle sur des réseaux informatiques sans fil ou sur des réseaux de téléphonie, etc.) ;
- Déterminer si le matériel ou le fichier examiné est un original, un original modifié ou une contrefaçon (documents numérisés, puis modifiés par ordinateur, cartes bancaires contrefaites, décodeurs de télévision numérique à péage « flashés », etc.) ;
- Déterminer à quoi sert le matériel ou logiciel examiné, dire s'il est fonctionnel et s'il a servi (ex. : dispositif électronique de piégeage de distributeur automatique de billets de type *skimmer*, application de type cheval de Troie ou *dialer*, etc.).

Notons que le contenu (numérique ou analogique) des objets analysés n'est pas le seul intérêt : le contenant ou support (physique) peut lui aussi avoir un intérêt. Ainsi, une marque, un modèle et un numéro de série imprimés, estampés, gravés ou embossés sur le contenant peuvent permettre d'identifier un composant électronique indéterminé, un téléphone GSM inconnu, ou un disque dur en panne : sans ces références, l'analyse de ces objets pourrait s'avérer impossible.

De même, on peut trouver sur le contenant des éléments de preuve « classiques » tels qu'ADN, empreintes digitales, résidus d'explosifs, produits contaminants de nature chimique ou bactériologique, etc. Par ailleurs, l'interface d'accès aux données peut être entravée par un obstacle physique qu'il convient au préalable de retirer par un pré-traitement. Tous ces cas obligent à déterminer un ordre d'intervention entre les spécialistes des différents domaines et à prendre des précautions particulières supplémentaires : l'intervenant N doit effectuer des traitements qui ne compromettent pas le succès des traitements de l'intervenant N+1 (ex. : manipulation d'un CD avec des gants pour préserver les empreintes digitales, décontamination chimique d'un téléphone GSM sans abîmer les circuits internes, attaques physico-chimiques sur une masse de polymères pour en dégager un composant électronique sans l'endommager, etc.).

L'analyse post mortem peut, au préalable, nécessiter l'emploi de techniques et outils utilisés en récupération de données : c'est notamment le cas quand le disque dur saisi est en panne (ex. : disque dur de vidéosurveillance dans des locaux incendiés, disque dur de « boîte noire » dans un navire qui a sombré, etc.), ou quand, au cours de la perquisition, le suspect a échappé momentanément à la vigilance de ses « visiteurs » et a endommagé volontairement des supports de stockage de données.

Contrairement aux autres activités décrites supra, l'expertise forensique a pour contrainte principale la non-modification absolue de la preuve : en effet, une éventuelle contre-expertise doit pouvoir, dans les mêmes conditions que la première expertise, retrouver les mêmes données. À l'issue de son travail, l'expert forensique doit donc restituer la preuve strictement dans le même état que celui dans lequel on la lui a fournie. En particulier, la machine et les outils d'analyse utilisés ne doivent pas modifier les fichiers journaux (*logs*) du système analysé **[27]** et les dates d'accès aux fichiers analysés ne doivent pas utiliser de mémoire d'échange (swap) sur les supports analysés **[28]** et ne doivent laisser aucune trace de montage sur les supports analysés **[29]**. L'utilisation de solutions de blocage en écriture (matériel **[30]** ou logiciel) et/ou la réalisation de copies parfaites (« images ») permettent de répondre à cette contrainte de préservation de l'original, avec une rigueur parfois poussée à l'extrême **[31]**.

L'une des difficultés majeures de l'analyse forensique réside généralement dans le fait que l'on ne sait pas forcément ce que l'on cherche exactement (« faire toute observation utile à la manifestation de la vérité ») : sauf les cas où l'utilisateur de l'objet saisi coopère en toute bonne foi (ce qui n'est pas fréquent… et encore faut-il qu'il ne s'agisse pas – par exemple – de l'ordinateur d'une personne assassinée !), l'expert forensique n'a aucune idée a priori du contenu précis et détaillé dudit objet. Et, sauf les cas où l'auteur de l'infraction est identifié avec une certitude absolue avant même l'analyse forensique, l'utilisateur de l'objet saisi n'est encore que simple suspect au moment de l'analyse : l'expert n'a donc aucune garantie quant à la présence ou non de ce qu'il recherche ! On le voit, la différence est majeure avec l'activité « récupération de données » décrite au début du présent article.

L'autre difficulté de l'analyse forensique est le principe théorique de l'exhaustivité de l'analyse. La preuve matérielle d'une infraction peut en effet reposer sur un seul et unique fichier, inclus dans un autre fichier, lui-même sous un format exotique, le tout stocké au fin fond d'un répertoire saugrenu. Ainsi un « simple consommateur » d'images pédopornographiques devient un violeur si, parmi les dizaines de milliers d'images trouvées sur son disque dur et issues d'Internet, on trouve une seule image « faite maison » par l'intéressé : si l'expert forensique se contente de signaler les 75.000 premières images, le suspect risque 2 années de prison ; si l'expert passe à côté de la 75.001ème image, le suspect échappe à 20 ans de prison. De même, au cours de l'analyse de l'ordinateur de la victime d'un assassinat, l'expert qui ne voit pas, au milieu de 1.200 emails, l'unique email de menaces reçu quelques jours avant le drame, permet à un criminel de couler des jours heureux en liberté.

Cette nécessité de l'exhaustivité, bien entendu, se heurte aux contraintes de délai d'analyse et de coût maximum imposées à l'expert forensique (toute heure travaillée étant facturée par l'expert), elles-mêmes imposées par la nécessité d'une justice aux délais et aux dépenses maîtrisés. Les outils de recherche automatisée de données existent et permettent d'accélérer certaines

analyses forensiques : parmi ceux-ci figurent naturellement la recherche de mots-clés après indexation de contenus et les comparaisons de fichiers par *hash* (pour élimination ou détection automatique). Ces outils atteignent cependant rapidement leurs limites : usage de l'argot, du verlan, du langage SMS et de « mots codés [32] » pour l'indexation, présence (par exemple) d'images pédopornographiques originales « faites maison » qui, évidemment, ne peuvent être référencées dans des bases de hash si elles ne sont pas encore connues. En tout état de cause, aussi performantes soient-elles, des solutions de recherche ou de tri automatiques ne peuvent garantir un résultat fiable à 100%, la présence de faux-positifs ou de faux-négatifs étant inévitable ; or, comme cité supra, la culpabilité d'un suspect peut reposer sur un seul et unique fichier.

Il n'est donc pas rare que l'expert forensique, aussi bon et aussi bien équipé soit-il, passe des journées entières à lire manuellement le contenu de centaines d'emails ou de fichiers textes un à un, et visualise une à une des milliers de vidéos : dès lors, comment s'étonner que l'analyse forensique d'un disque dur standard de 250 Go, qui peut contenir en données l'équivalent de toutes les collections physiques de la Bibliothèque Nationale de France, dure plusieurs semaines et coûte plusieurs milliers d'euros à la justice ?

Comme certaines procédures exigent des résultats dans les toutes premières heures suivant la saisie, en tous temps et en tous lieux, la quasi-totalité des administrations françaises a mis en place, formé et équipé des personnels d'intervention spécialisés, soit au niveau local, soit au niveau national, avec une capacité de projection rapide. La gendarmerie dispose ainsi de plus de 150 enquêteurs N-TECH spécialisés en technologies numériques répartis sur tout le territoire (en plus du laboratoire criminalistique central qu'est l'Institut de Recherche Criminelle – IRCGN), la Police nationale dispose de plusieurs dizaines d'enquêteurs spécialisés en criminalité informatique (ESCI) – en plus de l'Office Central de Lutte contre la Criminalité liée aux Technologies de l'Information et de la Communication (OCLCTIC) et du laboratoire d'Ecully, la Préfecture de Police de Paris dispose de la BEFTI et de la Brigade de Protection des Mineurs (BPM [33]), les douanes disposent du Centre de Recueil de la Preuve Informatique (CRPI), etc.

Conclusion

La récupération de données, l'e-discovery, l'intervention sur incident et l'analyse forensique ont toutes pour objet la recherche, l'extraction, voire la réparation et l'interprétation de données sur des réseaux ou supports de stockage. Malgré leurs différences d'approche et de contexte, elles doivent faire face à deux défis communs : d'une part, le compromis à trouver entre les délais [34] et la masse d'informations à traiter, d'autre part, l'équilibre à définir entre les moyens à mettre en œuvre pour arriver au résultat et le coût de cette mise en œuvre [35] rapporté à l'importance du résultat espéré.

Notes

[1] La « charte de bonnes pratiques relatives à la récupération de données contenues sur des supports informatiques endommagés » est disponible sur :

www.intelligence-economique.gouv.fr/IMG/pdf/Charte_de_bonnes_pratiques.pdf

[2] PhotoRec, PC *Inspector File Recovery, Easy Recovery*...

[3] CD/DVD rayé, bande DAT vrillée, disquette pliée...

[4] *Self-Monitoring, Analysis, and Reporting Technology*

[5] *Host Protected Area*

[6] *Device Configuration Overlay*

[7] Le lecteur en quête d'approfondissement pourra par exemple commencer par l'article *AT Attachment* de Wikipédia et par la présentation « Indiscrétions et 'zones constructeurs' des disques durs » de Laurent Dupuy (FreeSecurity) au *SSTIC* 2007, puis se plonger dans les versions successives de la norme ATA sur **www.t10.org/t13/project**.

[8] La P-List est la liste des secteurs défectueux en sortie de chaîne de production. Elle est de taille fixe. La G-List est la liste des secteurs qui deviennent défectueux au cours de la vie du disque dur. Elle grossit donc au fur et à mesure de « l'usure » du disque. La P-List et la G-List sont utilisées par le *sector remapper* dans le cadre du LBA (*Logical Block Addressing*).

[9] On citera notamment la solution PC 3000 d'ACE Laboratories (**www.acelab.ru**).

[10] La société prestataire demande alors au client de sérieuses garanties de bonne foi, quand elle ne refuse pas tout net la mission si les circonstances de la « perte » de la clé demeurent floues.

[11] 1 péta-octet = 1.000 téra-octets

[12] Encase version FIM (*Field Intelligence Model*) par Guidance Software et LiveWire par WetStone Technologies fonctionnent tous deux à partir de plateformes Microsoft Windows.

[13] Cette notion est ici abordée dans un sens très large.

[14] Article 323-1 et suivants du *Code pénal*

[15] Voir notamment l'édition 2005 du challenge DFRWS et ses résultats sur **www.dfrws.org**.

[16] Associé par exemple à des outils de prise d'empreinte (ex. : Tripwire), de *sniffing* réseau (ex. : Wireshark), d'antivirus/*antispyware*, de détection de *rootkit*, de détection d'intrusion (IDS), etc.

[17] Avec des outils tels que WinDASM, IDA Pro, etc.

[18] On utilise parfois l'expression anglaise « *law enforcement agencies* » (LEA).

[19] Notamment le *Forensic Information Technology Working Group* (FITWG) de l'ENFSI (*European Network of Forensic Science Institutes*, **www.enfsi.org**), l'AFSIN (Association Francophone des Spécialistes en Investigations Numériques, **www.afsin.org**) et diverses compagnies d'experts judiciaires (CEESD, CNEJITA, etc.)

[20] À ne pas confondre avec la criminologie, qui est la science humaine qui étudie d'un point de vue sociologique et statistique les auteurs, les victimes, les crimes, etc.

[21] Ah, la fameuse Fiat Uno blanche de Lady Diana !

[22] Ben Laden ou pas Ben Laden ?

[23] Direction de la Surveillance du Territoire (DST), Brigade d'Enquêtes aux Fraudes aux Technologies de l'Information (BEFTI) de la Préfecture de Police de Paris.

[24] Articles 323-1 et suivants du *Code pénal*, créés par la « loi Godfrain » de 1988.

[25] On utilise aussi l'expression « *live forensics* ».

[26] La distinction entre éléments matériels et immatériels est toutefois artificielle, puisque les données saisies le sont via stockage sur un support physique.

[27] La date de dernier démarrage de la machine analysée (par exemple) peut présenter un intérêt pour l'enquête.

[28] Quand le swap est pris sur les secteurs non alloués, il risque d'écraser des résidus de données effacées s'y trouvant déjà.

[29] Lorsqu'un système Windows monte des disques durs, il répartit automatiquement sa corbeille (\Recycler) et ses points de restauration (\System Volume Information) sur les disques montés. Quant aux « live CD Linux » dits « forensiques », nombreux sont ceux qui n'utilisent pas toutes les options de montage suivantes :

```
mount -ro -noswap -noatime -noexec -nodev.
```

[30] Tableau, Wiebetech, *Intelligent Computer Solutions* (ICS), Logicube, etc.

[31] À l'issue de leurs opérations, certains « imageurs » réactivent ainsi les fonctionnalités HPA et DCO qu'ils avaient désactivées pour réaliser la copie d'un disque dur. Le produit *Deepspar Disk Imager* va même jusqu'à désactiver les compteurs SMART et le *G-List remapper* du disque dur copié !

[32] Tels que les énigmatiques messages diffusés à la radio par les résistants français pendant la Seconde Guerre mondiale.

[33] Le « groupe cyber » de la BPM s'occupe de pédopornographie en lien avec les technologies numériques.

[34] Les données à récupérer sont critiques et le client ne peut s'en passer très longtemps, l'e-discovery et l'analyse forensique doivent obtenir des résultats rapides, l'intervention sur incident doit permettre un retour rapide à la normale.

[35] En temps, en argent, en hommes, etc.

Network Forensic : cherchez les traces sur le réseau

Packets o[n|f] the Storm

Il peut sembler étrange de parler d'analyse réseau dans un dossier consacré à l'analyse post mortem. Si l'on peut étudier le comportement d'une machine à travers ce qu'elle envoie ou reçoit, c'est que la « bête » est encore vivante. Dans certains cas cependant – en l'absence de logs par exemple – c'est pourtant une des seules choses à faire. Cet article présente une méthode – sans prétention – d'analyse de traces réseau à l'aide de TShark et d'outils OpenSource.

mots clés : ***analyse réseau / Wireshark / pcap***

Contexte

Qu'entend-on par « *network forensic* » ? Quels sont les cas où cette méthode est employée ?

Le terme est difficilement transposable en français. Il devrait se traduire en « analyse réseau post-mortem » ce qui n'a pas vraiment de sens : si une machine émet ou reçoit des paquets, elle n'est pas vraiment morte. Au pire pourrait-on qualifier son état de NDE (*Near Death Experience*).

Le « network forensic » est souvent défini comme une analyse de trafic en vue de résoudre ou déterminer l'origine d'incidents survenus sur un réseau. Cette définition, large, recouvre des cas très divers, allant de la mauvaise configuration d'un ordinateur ou d'un équipement réseau jusqu'à l'attaque DDoS en passant par toutes sortes d'anomalies. Dans le contexte qui nous intéresse, nous préciserons donc le périmètre dans lequel nous nous inscrivons en parlant « d'incidents de sécurité survenus sur un réseau ».

En résumé, il s'agit d'analyser le trafic entrant et sortant d'une machine et d'y rechercher toute trace d'intrusion ou d'activité illicite. Cela présente plusieurs avantages. Premièrement, il n'est rien de plus éphémère que le trafic réseau (à part la NVRAM peut-être). Le conserver sur disque permet de fixer des éléments potentiels de preuve. Ensuite, tout n'est pas forcément journalisé sur la machine compromise ou suspectée de l'être. Les traces réseau récoltées sont d'une aide précieuse pour « boucher les trous » dans les fichiers de logs. Enfin, aucune intervention sur la machine elle-même n'est nécessaire, elle restera donc en l'état avant, pendant et après l'analyse. Il y a peu de risque de détruire par mégarde des preuves ou d'effacer des traces en local.

Précisons que si l'on n'y prête pas suffisamment attention, le « network forensic » peut exposer celui (ou celle) qui le pratique sans précaution aux sanctions prévues par la loi dans le cadre de la protection des communications privées. Ce risque s'applique également à l'analyse du contenu d'un ordinateur, cela va de soi, mais l'analyse réseau peut facilement causer des dommages collatéraux comme l'interception de paquets non incriminés ou l'accès au contenu de conversations privées (courriels reçus et envoyés, chats ou dialogues IRC, sites visités, fichiers échangés, etc.). De l'analyse réseau à l'espionnage, ou à l'écoute illégale, il n'y a parfois que quelques paquets à ouvrir...

Il faut donc être très prudent et faire preuve de la plus grande rigueur, durant la collecte des traces ainsi qu'au cours de leur analyse. Une forte dose de déontologie n'est pas à exclure. La présence de tiers durant l'épluchage des paquets peut constituer une bonne couverture, puisque, suivant l'adage, « plus on est de mouillés, moins on risque de plonger seul ».

Allons maintenant droit au but et plantons le décor. Selon la formule consacrée, toute ressemblance dans ce qui va suivre avec des personnages existants ou ayant existé, ne serait que pure coïncidence (... ou pas [1]).

Nous sommes un lundi matin, le patron du service informatique vous convoque, l'heure est grave. On a découvert sur le réseau un PC au comportement suspicieux. Problème : l'administrateur de cet ordinateur est en congé et il n'est pas question de se connecter à cette machine avant son retour. Éteindre le poste est également exclu, il fait office de serveur de fichiers pour tout le service Comptabilité. Il faut rapidement savoir ce qui se passe car le D.I. lit la presse informatique, devient nerveux et voit des virus et des stagiaires chinois(e)s partout. Il est hors de question de laisser un *botnet* s'installer sur le réseau interne. Si un cheval de Troie est installé sur la machine, il n'est pas plus question de le laisser exfiltrer tous les fichiers. Si le comportement de la machine s'avère réellement anormal, elle sera isolée du réseau avant d'être auditée plus sérieusement, si besoin par les services de police compétents. Il s'agit donc, en premier lieu, d'effectuer une levée de doute sans hypothéquer les investigations futures.

Variante : nous sommes mardi matin et vous êtes averti qu'un utilisateur a malencontreusement cliqué sur un fichier téléchargé sur un site Web « bizarre ». Depuis, sa connexion réseau est étrangement lente et le disque dur n'arrête pas de brouter, ça sent le virus à plein nez. Sa machine a été déconnectée du réseau préventivement et, avant de se lancer dans une analyse du disque et des fichiers qui y sont installés, on voudrait bien savoir ce qui se passe quand on la reconnecte au réseau : des données sont-elles exfiltrées ? Si oui, lesquelles et vers où ? Le maliciel cherche t-il à infecter d'autres machines ?

Dans les deux cas, un passage par la case Analyse réseau s'impose.

La méthode

L'excellent Richard Bejtlich, auteur du non moins excellent *Tao of Network Security Monitoring*, propose une approche « *top to bottom* » de l'analyse de traces réseau.

Le tableau 1 présente cette approche dans ses grandes lignes et cite des outils qui répondent aux besoins identifiés. La terminologie est celle de R. Bejtlich.

La première étape de l'analyse consiste à produire à partir d'une capture des statistiques génériques : nombre de paquets recueillis, heure de début et de fin de la capture, etc.

Guillaume Arcas
guillaume.arcas@free.fr

Etape	Couche	Description	Outils
1	Statistical	Vue « haut niveau »	Capinfos
2	Session	Qui parle avec qui, quels protocoles utilisés, etc. ?	Argus, TShark, ntop, dnstop
3	Alert	Utilisation de signature pour identifier les paquets suspects.	Snort, Bro
4	Full content	Recherche d'anomalies	Tcpdump, TShark, Wireshark, ngrep, dnsdump

TI

A l'étape suivante, on s'intéresse aux conversations entre hôtes. Elle permet de répondre – ne serait-ce que partiellement – aux questions suivantes : qui parle avec qui, combien de machines différentes ont été impliquées dans les échanges, quels sont les protocoles utilisés, quelle a été la durée de ces conversations, quels volumes de données ont-ils été échangés ?

La troisième étape consiste à rechercher toute trace d'anomalie ou d'attaque dans les paquets à partir de signatures ou de règles.

Enfin, la dernière étape permet, à partir de l'examen détaillé du contenu de certains paquets, de confirmer les résultats obtenus aux étapes précédentes.

La boîte à outils

On ne devrait plus présenter la `libpcap` **[2]**, mais je vais quand même lui consacrer quelques lignes car tous les outils cités l'utilisent.

Cette bibliothèque fournit une interface indépendante du système d'exploitation, qui favorise la capture de paquets, y compris ceux non destinés à, ni émis par la machine hôte. Les paquets peuvent être stockés dans des fichiers dans un format compréhensible par tout utilitaire qui s'appuie sur les fonctions de la `libpcap`. Il est donc possible de lire des fichiers de traces générés sous Unix sur une machine Windows, par exemple, ou bien encore de traiter avec `Snort` un fichier créé par `tcpdump`.

Quoique perfectible et non exempte de vulnérabilités, la `libpcap` est un standard de facto pour la collecte de trafic réseau et la liste des outils liés à cette bibliothèque est longue comme un jour sans pain (ou sans RER B).

C'est pourquoi notre boîte à outils en est remplie. Qu'y trouve t-on ?

Tout d'abord, quelques utilitaires de collecte grâce auxquels les communications non chiffrées sur le réseau n'auront plus de secrets pour vous : `tcpdump`, `TShark`, `dumpcap`, `Snort`, pour n'en citer que quelques-uns. Leur rôle est simplissime : lire le trafic réseau depuis l'interface de la machine sur laquelle ils s'exécutent et le « dumper » dans des fichiers.

Ensuite, il nous faut des utilitaires pour restituer ce trafic en un format compréhensible par le commun des mortels. Le choix est varié, mais deux utilitaires sortent largement du lot : `Wireshark` et `TShark`. Ces deux outils offrent les mêmes fonctionnalités. `Wireshark` est utilisé en mode graphique. `TShark` est utilisé en ligne de commande et peut être appelé dans des scripts *shell*.

On pourrait presque s'arrêter à ces deux outils, mais on tirera également un très grand avantage de logiciels, comme Snort et Bro, pour leurs fonctionnalités de détection. Et puis, cela tombe bien, eux-aussi s'appuient sur la `libpcap`.

`Ntop` nous sera également d'une aide précieuse : cet outil dresse une cartographie des flux réseau sans entrer dans le contenu, ce qui est l'idéal pour une vision « macro » du trafic. Dans le même ordre d'idée, Argus apporte des informations similaires.

Enfin, la panoplie ne saurait être complète sans un outil d'anonymisation des paquets. Nous utiliserons pour cela `AnonTool`.

Enfin, n'oublions pas l'ami Google. Revenons donc maintenant à nos moutons.

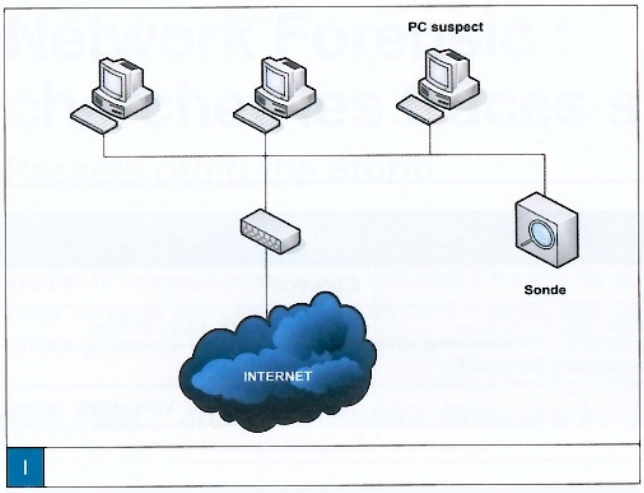
Où poser la sonde ?

Nous voici donc face à la machine suspecte. Première étape de l'analyse : collecter le plus de traces possibles émises et reçues par celle-ci sans interférer sur le trafic. Le positionnement de la sonde dans le réseau dépend donc des contraintes propres de l'architecture. Et la pertinence de l'analyse va quant à elle fortement dépendre des points de collecte retenus.

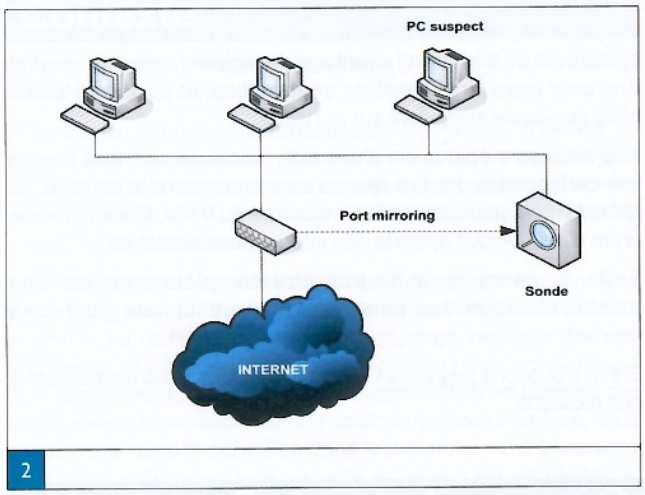
L'idéal est d'être au plus près de la machine, autant que faire se peut. On évitera ainsi les problèmes liés, entre autres, aux règles de translation d'adresses (NAT). On limitera également les risques de capturer du trafic « tiers » (c'est-à-dire sans rapport avec l'incident que l'on cherche à résoudre).

Dans la plupart des cas, la solution est simple : placer une machine sur le même segment réseau que l'ordinateur observé et utiliser un des utilitaires de collecte cités précédemment en plaçant la carte réseau en mode « *promiscuous* ». Dans ce mode, on capture tous les paquets qui passent par le brin réseau : le tour est (presque) joué, mais l'interaction avec les autres machines peut nuire à l'analyse (voir Figure 1, page suivante).

Cela peut se compliquer si la machine observée est isolée dans un VLAN. Il peut alors être nécessaire de modifier la configuration d'équipements réseau tels que les commutateurs (*switches*) et de faire de la recopie de ports (*port mirroring*). En gros, on configure l'équipement pour recopier le trafic d'un port (généralement associé à un VLAN) vers un autre sur lequel on branche la sonde. Généralement, cette solution n'est guère appréciée des techniciens réseau, ne serait-ce que parce qu'elle augmente la charge des équipements.

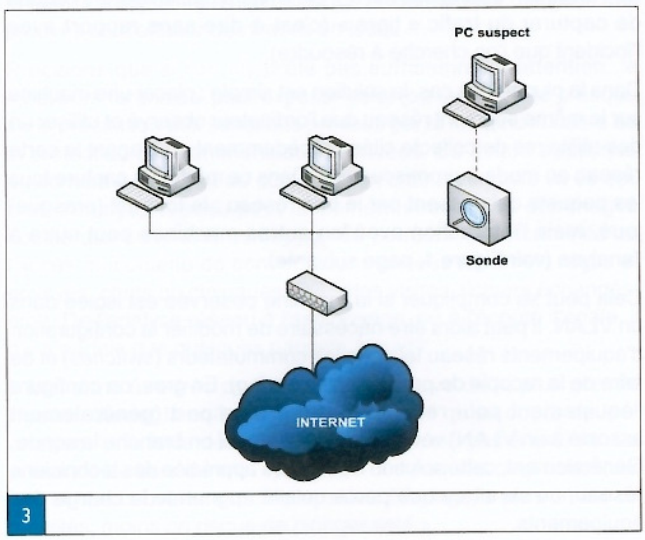


1



2

Autant dire que si l'on peut tout bonnement intercaler, entre l'ordinateur observé et le reste du réseau, une machine qui fera office de passerelle et de sonde, on est au paradis. D'une part, on isole le PC du reste du monde et du réseau interne, ensuite on est certain que ce qui en sort n'est pas « pollué » par du trafic légitime. Enfin, on peut « limiter la casse » en ne laissant passer que certains types de trafics sans que l'analyse en pâtisse. Dans le cas d'un botnet, on peut filtrer les charges utiles (SPAM ou attaques DDoS) tout en laissant passer les flux vers le canal de contrôle.



3

Cette configuration de sonde/passerelle permet d'allier l'utile à l'agréable : on y installera non seulement les outils destinés à collecter le trafic réseau, mais également quelques compléments tout aussi utiles pour déblayer le terrain : un serveur SMTP (PostFix pour ne citer qu'un exemple) permettra de *spooler* les courriels éventuellement émis par la machine auditée, un résolveur DNS (dnsmasq) de journaliser les requêtes, et, si l'on veut aller plus loin, un serveur DNS (Bind par exemple) permettra de rediriger les flux en répondant de manière contrôlée aux requêtes. On pourra ainsi, dans le cas d'un *keylogger*, modifier la destination des fichiers extraits par le *spyware* (si tant est que le site d'*upload* soit associé à un enregistrement DNS, bien sûr). Un bon vieux proxy Squid complètera la panoplie de base, sans oublier un serveur DHCP (les machines Windows en sont très friandes).

Point très important : l'horloge de cette passerelle – et d'une manière générale, celle de toutes les machines utilisées pour collecter des traces – doit être parfaitement synchronisée. NTP est votre ami.

L'objectif est d'étendre l'analyse et de ne pas se contenter des seuls paquets. Certes, tout ce qui se trouvera dans les logs des outils cités précédemment y sera aussi (dans les paquets). On peut donc penser que l'installation de tout cet attirail est superflue. Que nenni : si vous vous trouvez face à un « *mass-mailer* », vous n'allez tout de même pas le laisser *spammer* la planète sous prétexte que vous n'avez pas encore déterminé s'il s'agit d'un « simple » virus ou d'un botnet ? Il est donc tentant de bloquer tout flux sortant de la machine suspecte. Or, de très nombreux agents infectieux vérifient l'état de la connexion de leur hôte avant de lancer leur campagne de distribution de pourriels. De la même façon que « Pas de bras, pas de chocolat », « Pas d'Internet, pas d'activité ».

Si la machine intercalée entre le PC et l'Internet fait tourner un relais SMTP qui ne fera que spooler les courriels sans les relayer, l'agent aura cette impression – fausse en l'occurrence – qu'il peut atteindre Internet. Ensuite, il est beaucoup plus agréable de *grepper* le contenu de courriers dans des fichiers que de les reconstituer un par un à partir de paquets stockés, même si l'on pourrait tout aussi bien ngrepper **[3]** un fichier `pcap`. Et, au risque d'être lourd avec ça, grepper des messages sur un serveur « de prod' », c'est fourrer son nez dans le courrier des autres. Dans le cas d'un relais SMTP dédié, aucune chance de trouver autre chose que de la pub pour les petites pilules bleues qui font grandir ou les actions de la dernière start-up du coin.

Le raisonnement vaut pour les autres services de base comme le DNS. Ainsi outillé, non seulement vous mettez toutes les chances de votre côté, mais vous conservez aussi, ce qui n'est pas négligeable, le contrôle quasi total sur l'utilisation du réseau par l'ordinateur surveillé.

La collecte

Une fois la sonde en place, la collecte des paquets se fait à l'aide d'un banal `tcpdump` ou d'un `TShark`. `TShark` a cet avantage sur `tcpdump` d'être capable d'enregistrer le trafic en mode « *Ring Buffer* », c'est-à-dire en utilisant plusieurs fichiers. Ce mode est associé à une condition qui déclenche la création d'un nouveau fichier dès qu'elle est remplie. On peut ainsi décider de stocker les traces dans des fichiers de taille fixe, ou de créer des fichiers toutes les X minutes. Les fichiers ainsi créés pourront, si besoin, être fusionnés avec un outil comme `mergecap` (fourni avec `Wireshark`) ou `tcpslice`.

Pourquoi ne pas effectuer l'analyse en direct ?

Il faut être root pour capturer le trafic réseau sur la sonde et activer la carte en mode promiscuous. La libpcap n'est pas invulnérable. On a, dans un passé plus ou moins récent, vu des produits victimes de vulnérabilités dont certaines ont été habilement exploitées pour, comble du comble, compromettre des sondes de détection d'intrusion ! La remarque vaut aussi pour Wireshark, on utilisera donc un compte non privilégié pour l'analyse. Partir du principe que toute donnée capturée est potentiellement dangereuse est, plus qu'une sage précaution, une nécessité. N'oublions pas que le but poursuivi est de lever un doute ou qualifier en incident de sécurité une anomalie constatée sur le réseau, et non de fournir un nouveau *bot* à un quelconque pirate.

Notre sonde ne servira donc qu'à collecter les données qui seront analysées hors ligne sur une machine dédiée. On aura préalablement pris le soin de graver les traces, sur CD ou DVD, car elles peuvent se changer en « preuves » si l'analyse révèle un comportement ou un usage manifestement illicites de la machine auditée.

Soit un fichier pcap : qu'y a t-il dedans ?

Maintenant que nous avons récolté nos paquets, nous allons appliquer la méthode Betjlich.

Nous verrons à chaque étape que, comme en Perl, TIMTOWTDI [4] (voir encadré 1).

Analyse statistique

On effectue une première analyse statistique du trafic (répartition des flux par protocoles, nombre de paquets transmis, etc.). Cette étape n'apporte généralement pas grand chose, il faut bien le reconnaître, mais permet de planter le décor. L'utilitaire capinfos – compris dans le *package* Wireshark – affiche, à partir d'un fichier Pcap, les informations suivantes :

```
$ capinfos 20070828.pcap
File name: 20070828.pcap
File type: Wireshark/tcpdump/... - libpcap
Number of packets: 214799
File size: 16902553 bytes
Data size: 21359596 bytes
Capture duration: 805.533067 seconds
Start time: Tue Aug 28 14:42:08 2007
End time: Tue Aug 28 14:55:33 2007
Data rate: 26516.10 bytes/s
Data rate: 212128.81 bits/s
Average packet size: 99.44 bytes
```

Rien, donc, de bien transcendant.

Avec Tshark, nous obtenons ces statistiques un peu plus détaillées tout en restant très génériques :

```
$ tshark -r 20070828.pcap -z ptype,tree -q
===================================================================
 Port Type        value        rate        percent
-------------------------------------------------------------------
 Port Type       216004      0,572171
  UDP             97625      0,258598         45,20%
  TCP            118379      0,313573         54,80%
```

encadré 1

Des vertus de l'anonymat

Il est utile dans certains cas d'anonymiser les traces réseau.

Pourquoi ?

Tout d'abord parce que certaines concernent peut-être des hôtes « innocents » ou extérieurs à l'entreprise. Le but de l'analyse est, dans un premier temps tout du moins, de lever un doute. La connaissance des adresses IP réelles ne sera pas utile à ce stade. La seule qui nous intéresse est celle de la machine auscultée.

Ensuite parce qu'il peut être utile de partager des traces avec d'autres analystes. Un nettoyage de celles-ci s'imposera donc avant de faire appel à la bonté et à la générosité de la communauté. Le projet OpenPacket [5] propose ainsi des captures réseau à une communauté d'analystes. La bibliothèque de captures peut être enrichie par tout un chacun. On n'oubliera pas d'anonymiser le contenu des paquets et pas seulement les en-têtes.

Pour une raison un peu plus fine : l'analyse est menée par un être humain, qui peut se laisser influencer par ses a priori techniques. Le plus commun est d'associer un service ou un protocole à un numéro de port. Par exemple : « Port 80 = HTTP ». L'option -n de TShark affichera donc le numéro des ports et non le service qui leur est associé dans /etc/services. Cela ne garantit pas totalement d'une interprétation hâtive. Il convient donc de ne pas forcément croire tout ce qu'on va voir.

La plupart des outils effectuent des résolutions inverses pour afficher le nom d'une machine plutôt que son adresse IP. Tous proposent une option pour désactiver cette résolution, ce que l'on fera. Non pas, comme pourraient le croire – mais pas forcément à tort – certaines mauvaises langues, pour éviter d'afficher le nom des sites que visite son voisin de bureau, mais pour éviter toute lenteur à l'affichage due à la résolution (interrogation du DNS à chaque adresse IP). On garantira également la discrétion de l'analyse : quiconque aurait accès aux « logs » des résolveurs DNS à ce moment pourrait se douter de quelque chose ou, pire, accéder à des informations confidentielles. Enfin, on se prémunira des faux-semblants : le nom affiché au moment de l'analyse n'est peut-être pas ou plus celui qui était attribué quelques heures ou jours plus tôt à l'adresse IP (cas fréquent des adresses allouées dynamiquement). Il est donc prudent de ne travailler qu'à partir des adresses IP.

Dans notre cas, nous allons anonymiser les traces à l'aide d'anonymize_tool. Cet utilitaire lit les paquets contenus dans un fichier pcap, modifie les adresses IP et nettoie le contenu des paquets TCP et UDP. On obtient donc à la sortie un fichier pcap « propre » et, à cette étape, cela sera suffisant. L'option -a modifie les adresses IP, l'option -d le contenu des paquets, et l'option -c recalcule les *checksums* en conséquence. On peut aller jusqu'à nettoyer les adresses MAC si nécessaire.

```
$ ./anonymize_tool -f 20070828.pcap -a MAP -d HASH -c 20070828-anon.pcap
```

L'affichage de statistiques est déclenchée par l'option `-z`.

À ce stade, on note déjà une forte proportion de paquets UDP. A priori, ce protocole ne devrait pas se trouver dans une telle quantité dans un cas de navigation « classique » sur l'Internet.

Au passage, j'ouvre une autre parenthèse : ces statistiques réseau – volume du trafic entrant/sortant, part de chaque protocole dans ce trafic - devraient être collectées en permanence, et pas seulement en temps de crise. En dehors du fait que cela permet de produire de beaux graphes MRTG ou RRDTool qui agrémenteront les rapports hebdomadaires ou les bulletins de météo du réseau, ces données donneront un aperçu global de ce que recouvrent les expressions « trafic normal » ou « utilisation classique du réseau ». Ces informations ne sont pas confidentielles, ce serait bête de s'en priver. Et si vous n'aimez pas les graphes, gardez au moins les résultats dans un fichier. Avant de la parer, une crise se prépare. Je ferme la parenthèse.

Toujours avec l'option `-z` de `Tshark`, on peut visualiser la distribution des paquets par protocoles de manière encore plus fine :

```
$ tshark -r 20070828.pcap -z io,phs -q
===================================================================
Protocol Hierarchy Statistics
Filter: frame

frame                                    frames:214799 bytes:21359596
  eth                                    frames:214799 bytes:21359596
    ip                                   frames:214785 bytes:21358828
      udp                                frames:95185 bytes:8240818
        nbdgm                            frames:17 bytes:4184
          smb                            frames:17 bytes:4184
            mailslot                     frames:17 bytes:4184
              browser                    frames:17 bytes:4184
        nbns                             frames:162 bytes:15155
        data                             frames:72646 bytes:4633353
        edonkey                          frames:11 bytes:737
        llc                              frames:11 bytes:645
          data                           frames:7 bytes:469
          malformed                      frames:4 bytes:176
        dns                              frames:22319 bytes:3584935
      icmp                               frames:1223 bytes:108216
      tcp                                frames:118377 bytes:13010594
        data                             frames:197 bytes:157554
        smtp                             frames:68644 bytes:10010162
    arp                                  frames:14 bytes:768
===================================================================
```

Ces mêmes statistiques peuvent être obtenues graphiquement à l'aide d'un outil comme `ntop`.

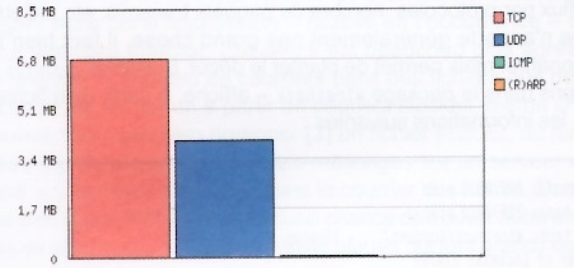
`ntop` est un utilitaire d'analyse de trafic qui offre un large panel de fonctionnalités : répartition des flux par protocoles, par adresses source et destination, statistiques de consommation de bande passante, suivi des conversations entre machines, etc. Ces données sont consultables depuis une interface Web, `ntop` possédant son propre moteur HTTP. Il peut capturer et traiter un trafic en « live » ou depuis un fichier `pcap`. C'est justement ce que nous allons faire :

```
$ /usr/sbin/ntop -f 20070828.pcap -m 10.0.0.0/255.0.0.0 -w 8888  --disable-instantsessionpurge -c
```

Les options `disable-instantsessionpurge` et `-c` modifient le comportement normal de `ntop` et désactivent les purges de sessions. `ntop` est en effet prévu pour un usage « au fil de l'eau » et ne garde pas indéfiniment les sessions expirées. Ces purges sont inutiles quand on lit un fichier `pcap`. L'option `-w` indique le port sur lequel le moteur HTTP de `ntop` est accessible à l'aide d'un navigateur.

`ntop` fournit les statistiques générales suivantes : voir Figures 4 et 5.

Protocol	Data	Percentage			
IP	10,9 MB	100,0%	TCP	6,8 MB	62,4%
			UDP	4,0 MB	37,0%
			ICMP	70,4 KB	0%
(R)ARP	0,4 KB	0%			



4 *Statistiques globales (ntop)*

TCP/UDP Protocol	Data	Flows	Accumulated Percentage / Historical Protocol View
DNS	2,0 MB	12,310	18,2%
NBios-IP	10,6 KB	91	0%
Mail	6,7 MB	2,649	61,4%
X11	2,4 KB	36	0%
eDonkey	1,6 KB	24	0%
BitTorrent	4,0 KB	62	0%
Messenger	7,8 KB	118	0%
Other TCP/UDP-based Protocols	2,2 MB	39,932	20,2%

5 *Répartition par protocoles applicatifs (ntop)*

Analyse des sessions

Passons à la couche « Session » que nous allons explorer, toujours à l'aide de `ntop` pour commencer.

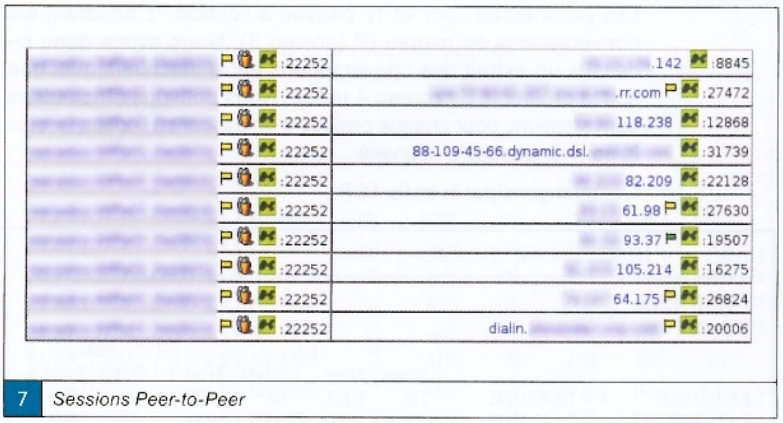
On distingue sur la capture ci-dessous deux pictogrammes à droite du nom des machines. L'enveloppe (1) indique un serveur SMTP. Le pictogramme vert (2) indique un pair P2P (voir Figure 6).

Regardons plus en détail les sessions P2P et SMTP : voir Figures 7 et 8.

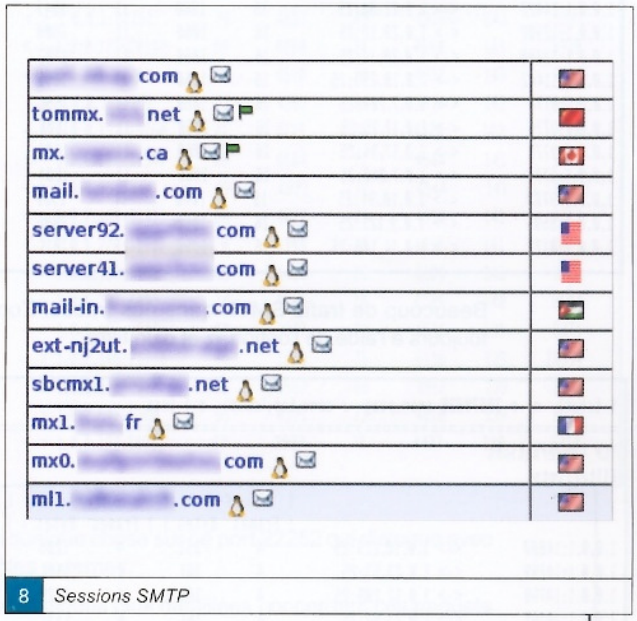
En quelques clics nous venons de mettre en évidence un trafic P2P et un trafic SMTP importants. Le trafic P2P semble provenir du port 22252 de la machine auditée. Les hôtes contactés sont nombreux et géographiquement bien répartis. Cette répartition est confirmée par le graphe suivant : voir Figure 9.

Ce type de trafic est-il normal et légitime pour un serveur de fichier ou un poste bureautique ? Gardons-nous de toute conclusion trop hâtive et continuons nos investigations. Revenons maintenant à `Tshark`, qui, lui aussi, permet d'afficher les conversations :

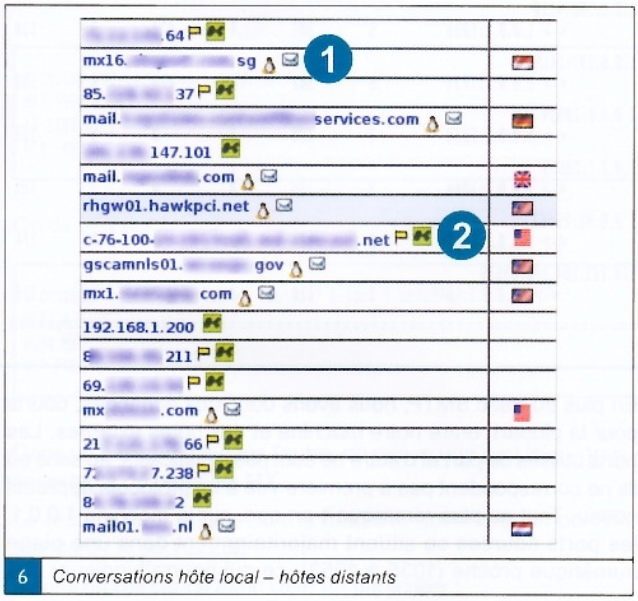
```
$ tshark -n -r 20070828.pcap -z conv,ip -q | less
================================================================================
IPv4 Conversations
Filter:<No Filter>
                                               |       <-      | |       ->      | |     Total     |
                                               | Frames  Bytes | | Frames  Bytes | | Frames  Bytes |
212.95.7.75          <-> 10.0.0.1                9260   700790    8904  2523011   18164  3223801
10.0.0.254           <-> 10.0.0.1                3002   224327    1503   172212    4505   396539
209.191.118.103      <-> 10.0.0.1                1570   242438    1855   128116    3425   370554
65.54.245.104        <-> 10.0.0.1                 792   122305     935    64602    1727   186907
216.39.53.1          <-> 10.0.0.1                 693   108941     820    56628    1513   165569
216.39.53.2          <-> 10.0.0.1                 649   100441     773    53306    1422   153747
66.196.97.250        <-> 10.0.0.1                 616    95537     729    50342    1345   145879
64.18.5.10           <-> 10.0.0.1                 594    91736     708    48816    1302   140552
68.142.202.247       <-> 10.0.0.1                 432    67137     510    35184     942   102321
209.191.88.239       <-> 10.0.0.1                 429    67479     510    35184     939   102663
65.54.244.168        <-> 10.0.0.1                 374    57756     447    30802     821    88558
64.18.6.14           <-> 10.0.0.1                 374    58144     445    30694     819    88838
216.39.53.3          <-> 10.0.0.1                 363    56137     432    29796     795    85933
209.191.88.247       <-> 10.0.0.1                 363    56917     431    29742     794    86659
65.54.244.72         <-> 10.0.0.1                 352    54189     419    28898     771    83087
64.18.7.10           <-> 10.0.0.1                 352    55111     418    28844     770    83955
65.54.244.232        <-> 10.0.0.1                 341    53702     403    27838     744    81540
```



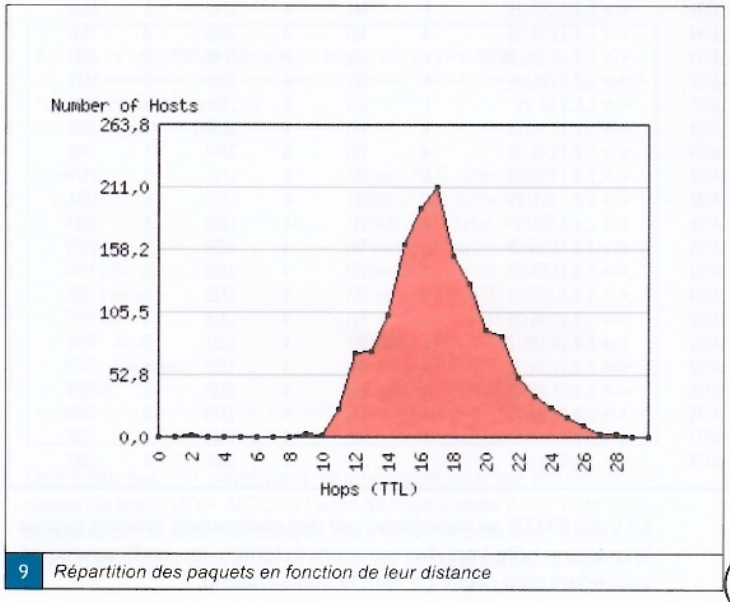
7 *Sessions Peer-to-Peer*



8 *Sessions SMTP*



6 *Conversations hôte local – hôtes distants*



9 *Répartition des paquets en fonction de leur distance*

Les paramètres `conv` et `ip` passés à l'option `-z` affichent les conversations au niveau IP (niveau 3). Nous avons donc ci-dessus un extrait des conversations entre machines quel que soit le protocole de niveau 4 utilisé. Nous pouvons afficher ces conversations pour chaque protocole en passant à l'option `-z conv` le nom du protocole souhaité.

Commençons par le trafic TCP :

```
$ tshark -n -r 20070828-anon.pcap -z conv,tcp -q | less
================================================================================
TCP Conversations
Filter:<No Filter>
                                               |  <-  | |  ->  | | Total |
                                               | Frames Bytes | | Frames Bytes | | Frames Bytes |
1.0.6.203:31632      <-> 1.0.0.1:1041            118    47853    136   113025    254   160878
1.0.0.1:14430        <-> 1.0.22.87:25             16     1060     11     1683     27     2743
1.0.0.1:14429        <-> 1.0.22.86:25             16     1060     11     1764     27     2824
1.0.0.1:14428        <-> 1.0.22.85:25             16     1060     11     1662     27     2722
1.0.0.1:14427        <-> 1.0.22.84:25             16     1060     11     1649     27     2709
1.0.0.1:12902        <-> 1.0.20.12:25             16     1060     11     1800     27     2860
1.0.0.1:12900        <-> 1.0.20.11:25             16     1060     11     1703     27     2763
1.0.0.1:11442        <-> 1.0.18.253:25            16     1060     11     1661     27     2721
1.0.0.1:9778         <-> 1.0.7.209:25             16     1060     11     1556     27     2616
1.0.0.1:9776         <-> 1.0.17.95:25             16     1060     11     1606     27     2666
1.0.0.1:9775         <-> 1.0.17.94:25             16     1060     11     1606     27     2666
1.0.0.1:9774         <-> 1.0.7.240:25             16     1060     11     1604     27     2664
1.0.0.1:9773         <-> 1.0.10.94:25             16     1060     11     1560     27     2620
1.0.0.1:8595         <-> 1.0.9.121:25             16     1060     11     1806     27     2866
1.0.0.1:8172         <-> 1.0.14.148:25            16     1060     11     1811     27     2871
```

Beaucoup de trafic SMTP dans nos traces. Zoomons un peu, toujours à l'aide de l'option `-z` et d'un filtre :

```
$ tshark -n -r 20070828-anon.pcap -z conv,tcp,'smtp' -q | less
================================================================================
TCP Conversations
Filter:smtp
                                               |  <-  | |  ->  | | Total |
                                               | Frames Bytes | | Frames Bytes | | Frames Bytes |
1.0.0.1:14800        <-> 1.0.10.229:25            4      361      4     1206      8     1567
1.0.0.1:14799        <-> 1.0.23.53:25             4      361      4     1191      8     1552
1.0.0.1:14798        <-> 1.0.12.209:25            4      361      4     1225      8     1586
1.0.0.1:14797        <-> 1.0.23.52:25             4      361      4     1331      8     1692
1.0.0.1:14796        <-> 1.0.9.78:25              4      361      4     1256      8     1617
1.0.0.1:14795        <-> 1.0.23.51:25             4      361      4     1253      8     1614
1.0.0.1:14794        <-> 1.0.23.50:25             4      361      4     1305      8     1666
1.0.0.1:14793        <-> 1.0.18.211:25            4      361      4     1202      8     1563
1.0.0.1:14792        <-> 1.0.7.209:25             4      361      4     1259      8     1620
1.0.0.1:14791        <-> 1.0.9.66:25              4      361      4     1324      8     1685
1.0.0.1:14790        <-> 1.0.23.49:25             4      361      4     1235      8     1596
1.0.0.1:14789        <-> 1.0.23.48:25             4      361      4     1247      8     1608
1.0.0.1:14788        <-> 1.0.23.47:25             4      361      4     1216      8     1577
1.0.0.1:14787        <-> 1.0.7.146:25             4      361      4     1293      8     1654
1.0.0.1:14786        <-> 1.0.9.78:25              4      361      4     1283      8     1644
1.0.0.1:14785        <-> 1.0.23.46:25             4      361      4     1259      8     1620
1.0.0.1:14783        <-> 1.0.21.34:25             4      361      4     1233      8     1594
1.0.0.1:14784        <-> 1.0.8.227:25             4      361      4     1225      8     1586
1.0.0.1:14782        <-> 1.0.23.45:25             4      361      4     1236      8     1597
1.0.0.1:14781        <-> 1.0.20.102:25            4      361      4     1231      8     1592
1.0.0.1:14780        <-> 1.0.23.44:25             4      361      4     1318      8     1679
1.0.0.1:14779        <-> 1.0.23.43:25             4      361      4     1239      8     1600
1.0.0.1:14778        <-> 1.0.23.42:25             4      361      4     1225      8     1586
1.0.0.1:14777        <-> 1.0.9.16:25              4      361      4     1344      8     1705
1.0.0.1:14776        <-> 1.0.9.78:25              4      361      4     1286      8     1647
```

Le trafic SMTP se caractérise par des connexions directes depuis la machine hôte vers des serveurs externes, les ports source se succèdent côté client, ça sent le mass-mailing.

Refaisons une passe en filtrant ce trafic :

```
$ tshark -n -r 20070828-anon.pcap -z conv,tcp,'!(tcp.port==25)' -q | less
================================================================================
TCP Conversations
Filter:!(tcp.port==25)
                                               |  <-  | |  ->  | | Total |
                                               | Frames Bytes | | Frames Bytes | | Frames Bytes |
1.0.6.203:31632
        <-> 1.0.0.1:1041                       118    47853    136   113025    254   160878
1.0.2.58:27995
        <-> 1.0.0.1:1034                         7      424      5      282     12      706
1.0.7.14:16385
        <-> 1.0.0.1:1053                         6      362      5      282     11      644
1.0.6.201:30161
        <-> 1.0.0.1:1039                         6      362      5      282     11      644
1.0.7.13:15120
        <-> 1.0.0.1:1052                         5      302      4      228      9      530
1.0.7.12:14858
        <-> 1.0.0.1:1051                         5      302      4      228      9      530
1.0.7.11:3351
        <-> 1.0.0.1:1050                         5      302      4      228      9      530
1.0.7.10:5650
        <-> 1.0.0.1:1049                         5      302      4      228      9      530
1.0.7.8:11547
        <-> 1.0.0.1:1047                         5      302      4      228      9      530
1.0.7.6:21716
        <-> 1.0.0.1:1045                         5      302      4      228      9      530
1.0.6.206:25246
        <-> 1.0.0.1:1044                         5      302      4      228      9      530
1.0.6.205:29020
        <-> 1.0.0.1:1043                         5      302      4      228      9      530
1.0.6.204:18332
        <-> 1.0.0.1:1042                         5      302      4      228      9      530
1.0.6.202:17650
        <-> 1.0.0.1:1040                         5      302      4      228      9      530
1.0.6.200:17144
        <-> 1.0.0.1:1038                         5      302      4      228      9      530
1.0.6.199:11950
        <-> 1.0.0.1:1037                         5      302      4      228      9      530
1.0.6.198:4110
        <-> 1.0.0.1:1036                         5      302      4      228      9      530
1.0.6.197:14185
        <-> 1.0.0.1:1035                         5      302      4      228      9      530
1.0.7.9:20524
        <-> 1.0.0.1:1048                         3      186      0        0      3      186
1.0.7.7:23539
        <-> 1.0.0.1:1046                         3      186      0        0      3      186
1.0.8.92:25662
        <-> 1.0.0.1:14090                        2      124      0        0      2      124
116.121.164.176:23539
        <-> 10.0.0.1:1046                        1      110      0        0      1      110
================================================================================
```

En plus du trafic SMTP, nous avons donc des échanges, courts pour la plupart, entre notre machine et des hôtes externes. Les ports utilisés de part et d'autre ne sont pas significatifs, au sens où ils ne correspondent pas à première vue à un protocole applicatif connu. Tout au plus remarque-t-on que, sur la machine 1.0.0.1, les ports sources se situent majoritairement dans une plage numérique proche (1035 à 1053), ce qui pourrait indiquer des

connexions faites par un même logiciel client vers des serveurs distants. Le volume des paquets échangés est aussi très proche pour de nombreuses connexions.

A noter que TShark affiche ces résultats par volume décroissant de données échangées.

Passons maintenant au trafic UDP :

```
$ tshark -n -r 20070828-anon.pcap -z conv,udp -q | less
================================================================================
UDP Conversations
Filter:<No Filter>
                                       |       <-      |  |       ->      |  |     Total     |
                                       | Frames  Bytes |  | Frames  Bytes |  | Frames  Bytes |
1.0.0.1:1059      <-> 1.0.3.235:53         242    28043     411    30966     653    59009
1.0.0.1:1063      <-> 1.0.3.235:53         224    26354     384    29261     608    55615
1.0.0.1:1056      <-> 1.0.3.235:53         207    24927     397    29833     604    54760
1.0.0.1:1061      <-> 1.0.3.235:53         184    22574     385    29075     569    51649
1.0.0.1:1025      <-> 1.0.3.235:53         135    15412     388    28924     523    44336
1.0.0.1:1057      <-> 1.0.3.235:53         148    17875     338    24963     486    42838
1.0.0.1:22252     <-> 1.0.6.95:10020       195    16202      31     2069     226    18271
1.0.0.1:22252     <-> 1.0.6.120:16235      183    15544      30     2002     213    17546
1.0.0.1:1058      <-> 1.0.3.235:53           0        0     202    14590     202    14590
1.0.0.1:1055      <-> 1.0.3.235:53           0        0     183    13303     183    13303
1.0.1.227:28145   <-> 1.0.0.1:22252         70     4409      96     7054     166    11463
1.0.0.1:1060      <-> 1.0.3.235:53           2      236     162    12038     164    12274
1.0.0.1:1062      <-> 1.0.3.235:53          11     1386     152    11374     163    12760
1.0.0.1:22252     <-> 1.0.3.230:16651       91     6623      70     4409     161    11032
1.0.0.1:22252     <-> 1.0.5.204:11217       73     4456      77     6926     150    11382
1.0.0.1:22252     <-> 1.0.2.248:4063        73     4433      76     4784     149     9217
1.0.0.1:22252     <-> 1.0.2.104:16951       72     4573      75     4717     147     9290
1.0.0.1:22252     <-> 1.0.2.222:3085        72     4527      74     4650     146     9177
1.0.0.1:22252     <-> 1.0.1.253:19087       71     4205      74     5452     145     9657
1.0.0.1:22252     <-> 1.0.3.225:1301        73     4479      71     4463     144     8942
```

Le trafic UDP se partage entre le protocole DNS, ce qui est logique compte tenu du trafic SMTP mis en évidence précédemment, et « autre chose ».

La typologie du trafic DNS peut être mise en évidence à l'aide de dnstop :

```
$ dnstop 20070828.pcap | grep -E '^.+\?'
MX? 6598 53.8
A? 5207 42.5
PTR? 457 3.7
```

Ou de dnsdump :

```
$ dnsdump -r 20070828.pcap -q "%qtype" | sort | uniq -c
5207 A
6598 MX
457 PTR
```

Dans un cas comme dans l'autre, on relève une part très – trop – importante de requêtes MX.

Le reste a pour origine, sur la machine observée, un port non standard : le port 22252.

Jetons un œil sur ce trafic UDP en particulier :

```
$ tshark -n -r 20070828-anon.pcap -z conv,udp,'!dns' -q | less
================================================================================
UDP Conversations
Filter:!dns
                                       |       <-      |  |       ->      |  |     Total     |
                                       | Frames  Bytes |  | Frames  Bytes |  | Frames  Bytes |
1.0.0.1:22252     <-> 1.0.6.95:10020       195    16202      31     2069     226    18271
1.0.0.1:22252     <-> 1.0.6.120:16235      183    15544      30     2002     213    17546
1.0.1.227:28145   <-> 1.0.0.1:22252         70     4409      96     7054     166    11463
1.0.0.1:22252     <-> 1.0.3.230:16651       91     6623      70     4409     161    11032
1.0.0.1:22252     <-> 1.0.5.204:11217       73     4456      77     6926     150    11382
1.0.0.1:22252     <-> 1.0.2.248:4063        73     4433      76     4784     149     9217
1.0.0.1:22252     <-> 1.0.2.104:16951       72     4573      75     4717     147     9290
1.0.0.1:22252     <-> 1.0.2.222:3085        72     4527      74     4650     146     9177
1.0.0.1:22252     <-> 1.0.1.253:19087       71     4205      74     5452     145     9657
1.0.0.1:22252     <-> 1.0.3.225:1301        73     4479      71     4463     144     8942
1.0.2.80:30877    <-> 1.0.0.1:22252         74     4657      69     4326     143     8983
1.0.0.1:22252     <-> 1.0.0.172:20142       69     4280      74     4650     143     8930
1.0.0.135:24311   <-> 1.0.0.1:22252         73     4590      70     4832     143     9422
1.0.2.207:32344   <-> 1.0.0.1:22252         73     4597      69     4719     142     9316
1.0.0.1:22252     <-> 1.0.0.132:15180       71     4715      71     4470     142     9185
1.0.0.1:22252     <-> 1.0.6.188:11547       69     4604      72     4544     141     9148
1.0.0.1:22252     <-> 1.0.3.93:3844         69     4002      72     4544     141     8546
1.0.0.1:22252     <-> 1.0.2.168:5191        68     4027      73     4590     141     8617
1.0.0.1:22252     <-> 1.0.1.220:11217       70     4393      71     4456     141     8849
1.0.0.1:22252     <-> 1.0.0.133:1181        68     4491      73     4604     141     9095
1.0.0.1:22252     <-> 1.0.1.237:4093        70     4115      70     4389     140     8504
1.0.0.1:22252     <-> 1.0.5.127:4046        67     3937      72     4537     139     8474
1.0.0.1:22252     <-> 1.0.2.148:21776       68     3958      71     6136     139    10094
1.0.2.126:32777   <-> 1.0.0.1:22252         73     4583      66     3962     139     8545
1.0.0.1:22252     <-> 1.0.2.64:15360        66     4311      73     4618     139     8929
1.0.0.1:22252     <-> 1.0.4.122:16951       67     3937      71     4477     138     8414
```

On a donc bien quelque chose sur ce port 22252 qui dialogue avec tout un tas d'hôtes externes.

Affichons des statistiques plus détaillées concernant ces paquets UDP :

```
$ tshark -n -r 20070828-anon.pcap -z io,phs,'udp.port == 22252' -q
================================================================================
Protocol Hierarchy Statistics
Filter: udp.port == 22252

frame                                    frames:73531 bytes:4705925
  eth                                    frames:73531 bytes:4705925
    ip                                   frames:73531 bytes:4705925
      udp                                frames:72687 bytes:4635824
        data                             frames:72646 bytes:4633353
        edonkey                          frames:11 bytes:737
        llc                              frames:11 bytes:645
          data                           frames:7 bytes:469
          malformed                      frames:4 bytes:176
      icmp                               frames:844 bytes:70101
================================================================================
```

Une ligne ressort nettement, qui nous dit que sur les 4635824 octets de trafic UDP, 4633353 sont de type « data ». On note aussi la présence de datagrammes eDonkey, et là, cela devient intéressant. Comment TShark a t-il détecté ce trafic ? À l'aide de ses dissecteurs

protocolaires, sortes de *wizards* internes qui permettent à TShark de remonter un peu plus haut dans les couches du modèle OSI.

Examinons donc ces paquets eDonkey :

```
$ tshark -n -r 20070828-anon.pcap -z conv,udp,'edonkey' -q | less
UDP Conversations
Filter:edonkey
                                          |       <-      | |       ->      | |     Total     |
                                          | Frames  Bytes | | Frames  Bytes | | Frames  Bytes |
1.0.0.1:22252       <-> 1.0.1.80:4665           0      0        4     268        4     268
1.0.0.1:22252       <-> 1.0.0.94:4665           0      0        3     201        3     201
1.0.0.1:22252       <-> 1.0.1.185:4665          0      0        2     134        2     134
1.0.0.1:22252       <-> 1.0.1.152:4665          0      0        2     134        2     134
```

On retrouve une donnée déjà vue précédemment : le port source 22252. On savait que « quelque chose » utilise ce port sur la machine, on sait maintenant que, pour quelques paquets tout du moins, ce « quelque chose » émet du trafic eDonkey. Il est donc très probable que ce « quelque chose » soit un client P2P.

Avant de passer à l'étape suivante, on notera qu'avec ntop, on obtient en quelques clics seulement ce qui nous a pris plusieurs commandes avec TShark.

Analyse des alertes

La troisième étape de notre analyse nous conduit à utiliser un IDS : Snort. Comme les outils précédents, Snort est capable de travailler à partir d'un fichier pcap. Pour tirer le meilleur parti de Snort, il est très conseillé d'utiliser les signatures produites par le projet Snort, mais aussi celles fournies par le projet *Bleeding Threats* **[6]**. Ces dernières sont plus orientées *malware* que les signatures Snort. Si un ver ou un virus est à l'action sur la machine, les signatures Bleeding ne devraient pas le rater.

Une fois les signatures en place, on lance Snort :

```
$ snort -r 20070828.pcap
```

Je vous fais grâce des lignes de *debug* affichées lors de la phase d'initialisation du moteur pour aller à l'essentiel :

```
Snort processed 214799 packets.
===================================================================
Breakdown by protocol:
    TCP: 118377     (55.111%)
    UDP: 95185      (44.314%)
   ICMP: 1223       (0.569%)
    ARP: 14         (0.007%)
  EAPOL: 0          (0.000%)
   IPv6: 0          (0.000%)
ETHLOOP: 0          (0.000%)
    IPX: 0          (0.000%)
   FRAG: 0          (0.000%)
  OTHER: 0          (0.000%)
DISCARD: 0          (0.000%)
===================================================================
Action Stats:
ALERTS: 54039
LOGGED: 54039
PASSED: 0
===================================================================
```

54.039 alertes ont été remontées par le moteur. Sur 214.799 paquets traités, ça fait un peu beaucoup, trop en tout cas pour être honnête !

Voici quelques alertes que Snort a généré à partir de notre fichier :

```
08/28-13:43:14.397796  [**] [1:2003310:1] BLEEDING-EDGE P2P Edonkey Publicize
                       File [**] [Classification: Potential Corporate Privacy
                       Violation] [Priority: 1] {UDP} 10.0.0.1:22252 ->
                       81.2.xxx.136:5298
08/28-13:43:14.397828  [**] [1:2003310:1] BLEEDING-EDGE P2P Edonkey Publicize
                       File [**] [Classification: Potential Corporate Privacy
                       Violation] [Priority: 1] {UDP} 10.0.0.1:22252 ->
                       81.57.135.xxx:9016
08/28-13:43:24.410938  [**] [1:2007634:1] BLEEDING-EDGE TROJAN Storm Worm
                       Encrypted Traffic Outbound - Likely Search by md5
                       [**] [Classification: A Network Trojan was detected]
                       [Priority: 1] {UDP} 10.0.0.1:22252 -> 81.88.xxx.121:53436
08/28-13:43:34.556414  [**] [1:2007635:1] BLEEDING-EDGE TROJAN Storm Worm
                       Encrypted Traffic Inbound - Likely Connect Ack [**]
                       [Classification: A Network Trojan was detected]
                       [Priority: 1] {UDP} 81.xxx.12.96:11827 -> 10.0.0.1:22252
```

On retrouve tous les éléments précédemment mis en lumière, mais cette fois, nous pouvons mettre un nom sur l'origine du phénomène : Storm Worm.

Un petit tour par Google – ou par le site des principaux éditeurs d'antivirus – nous apprend que Storm Worm est un malware de type botnet, qui utilise comme canal de contrôle un réseau P2P de type eDonkey/OverNet, et que la bestiole possède, parmi ses charges utiles, une belle capacité à spammer.

Pour valider ce dernier point, nous allons devoir inspecter les paquets, ce qui correspond à ce que Richard Bejtilch appelle les « *Full content data* ».

Analyse des paquets

Comme des courriels sortent de la machine, ils sont sauvegardés sur la sonde. Il est donc tentant de voir ce qu'il y a dedans.

note

Note : ces messages sont également contenus dans les traces collectées. Si l'on a juste ces traces, Wireshark permet de les reconstituer à partir du fichier pcap.

Voici ce qu'on y trouve en grandes quantités :

```
Subject: Here Ya Go
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 7bit

Hot News Finally Hits On ERMX, And It Was Worth The Wait.

EntreMetrix Inc. (ERMX)
$0.089 UP 11.25%

Gladiator challenge is now an ERMX asset. It is one of the biggest
promoters of mixed martial arts. This asset expansion will develop new
dividend opportunities. Review the release, and be ready for open on
Tuesday.
```

Voilà qui confirme la présence de Storm Worm sur la machine.

Une autre façon de procéder permet, dans ce cas, de certifier que le trafic SMTP sortant est du « spam » sans même avoir à lire les messages collectés. Elle consiste à compter le nombre de domaines uniques des adresses de messagerie source suivant la commande SMTP `MAIL FROM:`.

```
$ ngrep -W byline -P '>' -i '^MAIL FROM:' -I 20070828.pcap \
> tcp and dst port 25 and src host 10.0.0.1 \
> | grep -i '^mail from:' \
> | sed -r 's/^.*@([^>]+)>.*$/\1/' \
> | sort | uniq | wc -l
18832
```

Même sur la plupart des serveurs de messagerie, un tel nombre de domaines source différents pour des messages sortants serait suspect. Alors sur un serveur de fichiers ou un poste de travail...

Toujours à l'aide de `ngrep`, on peut lister les adresses utilisées dans le champ `From` :

```
$ ngrep -W byline -P '>' -i '^MAIL FROM:' -I 20070828.pcap | grep -i '^mail
from:' | cut -d: -f2 | sort -u
(...)
<utumtdae@xxxxx.com>>
<utumtdae@rrrrrrn.com>>
<utumtdae@fffffu.edu>>
<utumtdae@ttttter.com>>
<utumtdae@hhhhh.co.za>>
<utumtdae@sssss.com>>
<vanica@ceeeee.org>>
<vanica@kuuuuus.com>>
<vanica@tfdfdfdfdt.br>>
<vanica@rttti.edu>>
<vanica@wsdsdsdsda.com>>
<vanica@zzzzz.co.uk>>
<vermpy@ooooo.com>>
<vermpy@chhhhht.com.br>>
<vermpy@llllll.edu.au>>
<vermpy@ppppppd.com>>
(...)
```

La construction de ces adresses semble obéir à une règle ou provenir d'un dictionnaire.

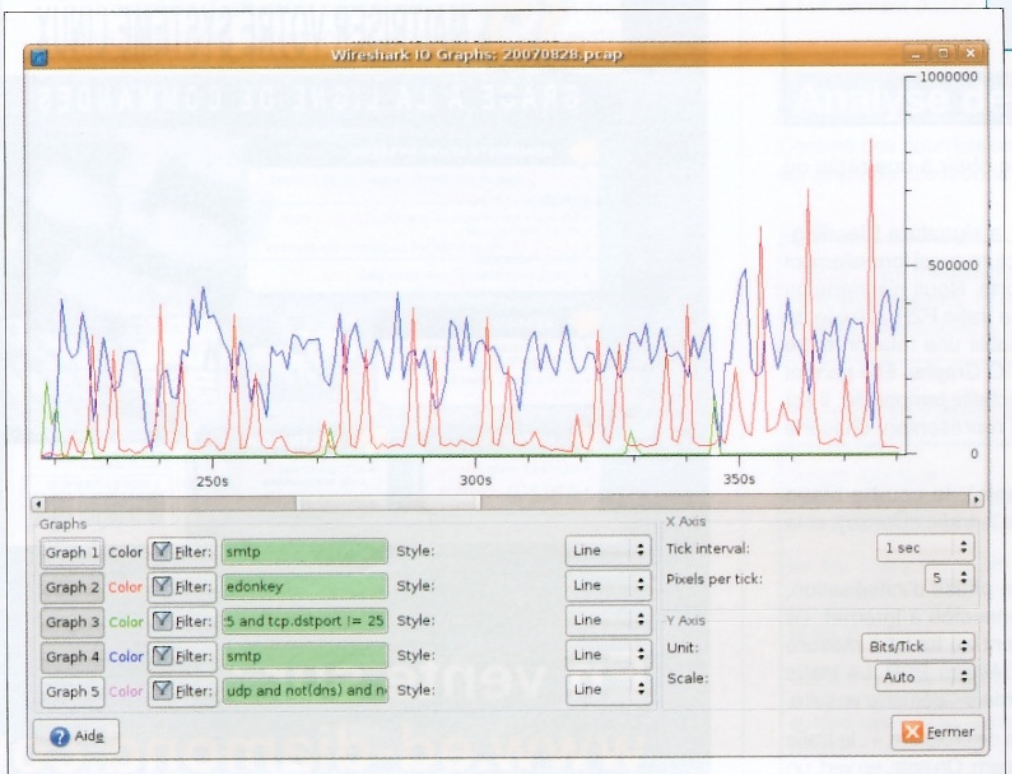
J'ouvre une dernière parenthèse : sans la signature Bleeding-Edge, il n'aurait pas été aussi facile de conclure aussi formellement à l'infection de la machine par Storm Worm. Nous n'aurions pu que constater les émissions de Spam et le trafic P2P. A l'aide de `Wireshark`, nous aurions cependant pu établir une relation entre ces deux trafics grâce à la fonctionnalité IO Graphs. Elle permet de visualiser l'évolution du trafic sur une échelle temporelle. Il est possible de superposer des courbes qui représentent chacune un type de flux.

Sur les schémas 10 et 11 (page suivante), la courbe bleue représente le trafic SMTP, la courbe rouge le trafic eDonkey, et la verte le trafic TCP hors SMTP.

La figure 10 représente le trafic lors de la phase d'initialisation, c'est-à-dire lorsque la machine est reconnectée à Internet. Le trafic eDonkey démarre assez rapidement au fur et à mesure que la machine contacte les pairs du réseau P2P. Le trafic SMTP – qui correspond aux envois de pourriels – démarre ensuite.

La machine atteint rapidement sa « vitesse de croisière » : le trafic P2P est permanent ainsi que les envois de Spam. On note, en vert, un trafic TCP, moins régulier et moins volumineux en dehors de quelques pointes qui semblent cependant précéder une reprise du Spam.

10 *Initialisation*



11 *Trafic en vitesse de croisière*

Conclusion

Nous voici au bout de notre analyse, mais pas au bout de nos peines. Nous pouvons dire que la machine est infectée – de manière sûre si on a eu la chance de disposer des signatures à jour ou de manière très probable sinon. Une analyse plus poussée de la machine et des programmes qui y sont installés devra la compléter. Les éléments recueillis à partir des traces réseau orienteront les recherches sur la machine d'une part, mais « légitiment » aussi cette analyse.

Notes

[1] Suivant une autre expression tout aussi consacrée ! :-)

[2] Disponible à l'adresse suivante : **http://www.tcpdump.org/**

[3] Ngrepper (v. tr.) : rechercher du texte dans un fichier pcap à l'aide d'expressions régulières et de l'utilitaire ngrep (**http://ngrep.sourceforge.net/**).

[4] « *There Is More Than One Way To Do It* » : il y a plusieurs façons de le faire.

[5] Projet OpenPacket : **http://openpacket.blogspot.com/**.

[6] **http://www.bleedingthreats.net/**

Liens

- TShark/Wireshark : **www.wireshark.org**
- Richard Bejtlich Blog : **taosecurity.blogspot.com**
- Anontool : **www.ics.forth.gr/dcs/Activities/Projects/anontool.html**
- ntop : www.ntop.org
- tcpslice : **www.tcpdump.org/related.html**
- dnstop : **dns.measurement-factory.com/tools/dnstop/**
- dnsdump : **dns.measurement-factory.com/tools/dnsdump/**

Remerciements

Un grand merci à David Lesperon qui saura pourquoi. :-)

3/5

Reconstruisez l'image d'un disque

Image d'un disque

Téléphones, ordinateurs, CD, DVD, disques durs, clés USB, baladeurs MP3, appareils photo numériques, nous sommes entourés de supports d'informations numériques. Confrontés à une perte de données, il peut être risqué de tenter une réparation sans avoir pris soin d'établir au préalable une copie. Cet article présente quelques utilisations de copies de supports numériques et donne quelques conseils pour les réaliser.

mots clés : *récupération de données / forensique / autopsie / image / disque / Linux*

Pourquoi réaliser une ou plusieurs copies ?

Tester plusieurs techniques de récupération

Les outils de réparation de systèmes de fichiers ne sont pas capables de gérer efficacement tous les problèmes. Cela ne va pas étonner grand monde. Pire, j'ai fait moi-même l'expérience de programmes de « réparation » supprimant tout élément considéré comme endommagé. À la fin de l'opération, le système de fichier était, il est vrai, consistant, mais désespérément vide. Il faut donc être en mesure de pouvoir tester plusieurs outils afin de trouver celui qui va réussir enfin à récupérer ses données sans risquer de tout perdre. La difficulté est de revenir à l'état initial. Certains outils disposent d'une fonction `Undo` pour annuler une réparation, mais celle-ci ne fonctionne pas si l'outil plante en plein milieu d'une opération ou si le système a été modifié depuis.

Éviter les erreurs de manipulation

Même en utilisant des techniques de récupération de données copiant les fichiers, nul n'est à l'abri d'erreurs de manipulation. Il vaut mieux prendre des précautions pour ne pas risquer d'altérer le média et empêcher toute récupération ultérieure. Une erreur humaine est souvent à l'origine de la perte de données. En état de stress, une erreur supplémentaire de manipulation n'est pas exclue.

Panne matérielle

En cas de panne matérielle légère et avec un peu de chance cependant, un disque dur reste malgré tout globalement accessible. Deux stratégies sont généralement à envisager : confier le disque à une société spécialisée ayant une expertise dans la récupération de données liée à des problèmes matériels ou bien prendre la situation en main. Considérant que la panne risque de se généraliser du fait même de l'utilisation du disque, il est préconisé de copier les fichiers les plus importants en premier, puis de dupliquer le disque tant que cela est possible.

Libérer la ressource

Lorsque des fichiers ont été effacés et qu'il s'agit de les récupérer, il faut absolument éviter d'écrire la moindre donnée sur le disque, sous peine de risquer d'écraser les fichiers perdus par de nouveaux ou bien d'écraser des blocs de données lorsque la taille de certains fichiers s'accroît. Les blocs de données d'un fichier effacé étant considérés comme libres, ils sont susceptibles d'être réalloués. Au moment de cette réallocation, le contenu du bloc « libre » est réinitialisé à 0 par le système d'exploitation, pour empêcher la lecture des données précédemment stockées. L'application peut alors y écrire de nouvelles données. Dès que la perte de fichier est détectée, pour éviter ce problème d'écrasement de données, il ne faut plus du tout utiliser le disque jusqu'à ce que toutes les données soient récupérées. En pratique, il peut y avoir une forte contrainte de temps rendant impossible de devoir immobiliser cette ressource : besoin d'utiliser des programmes présents sur ce disque, d'accéder à des données et divers autres impératifs de production. Une solution est de dupliquer le disque original et de le remettre en production dès la fin de la copie, le processus de récupération de données, parfois long et complexe, s'effectue alors dans un second temps. On terminera en réintégrant les fichiers récupérés.

Droit : la copie est mentionnée

Dans certains cas, il est indispensable de conserver une information en l'état où elle était à un instant donné. On pensera particulièrement à la recherche de preuves informatiques.

Voici l'article L. 332-4 du *Code de la propriété intellectuelle* (CPI) :

> En matière de logiciels et de bases de données, la saisie-contrefaçon est exécutée en vertu d'une ordonnance rendue sur requête par le Président du Tribunal de Grande Instance. Le Président autorise, s'il y a lieu, la saisie réelle. L'huissier instrumentaire ou le commissaire de police peut être assisté d'un expert désigné par le requérant. À défaut d'assignation ou de citation dans la quinzaine de la saisie, la saisie-contrefaçon est nulle. En outre, les commissaires de police sont tenus, à la demande de tout titulaire de droits sur un logiciel ou sur une base de données, d'opérer une saisie-description du logiciel ou de la base de données contrefaisant, saisie-description qui peut se concrétiser par une copie.

Une saisie-contrefaçon vise à démontrer une contrefaçon dans le cadre d'un procès. Dans ce cadre légal, comme le montre l'article de loi, la saisie des supports informatiques n'est pas l'unique possibilité pour acquérir des preuves, il est possible de réaliser des copies. En fait, il est plus facile d'obtenir le droit de réaliser une copie qu'une saisie, une saisie réelle dépossédant le saisi d'une partie de son matériel informatique, ce genre de saisie doit être particulièrement justifié. Je n'ai trouvé aucun texte sur les modalités pratiques de la copie, mais il semble nécessaire de prendre des précautions afin qu'une contre-expertise soit possible.

Christophe Grenier
grenier@cgsecurity.org
RSSI chez Global Service Provider

Dans le cadre d'une procédure pénale, là encore, une copie est possible :

Extrait de l'article 97 du *Code de procédure pénale* :

> Il est procédé à la saisie des données informatiques nécessaires à la manifestation de la vérité en plaçant sous main de justice soit le support physique de ces données, soit une copie réalisée en présence des personnes qui assistent à la perquisition.
>
> ...
>
> Si les nécessités de l'instruction ne s'y opposent pas, copie ou photocopie des documents ou des données informatiques placés sous main de justice peuvent être délivrées à leurs frais, dans le plus bref délai, aux intéressés qui en font la demande.

Nouveau Code de procédure civile, Article 145 :

> S'il existe un motif légitime de conserver ou d'établir avant tout procès la preuve de faits dont pourrait dépendre la solution d'un litige, les mesures d'instruction légalement admissibles peuvent être ordonnées à la demande de tout intéressé, sur requête ou en référé.

Article 493 :

> L'ordonnance sur requête est une décision provisoire rendue non contradictoirement dans les cas où le requérant est fondé à ne pas appeler de partie adverse.

Cela permet de ne pas prévenir la partie adverse si l'on craint que des informations soient détruites, mais attention, la saisie peut être contestée aussi bien sur la forme que le fond, et rendre la saisie nulle.

Article 812 :

> Le Président du tribunal est saisi par requête dans les cas spécifiés par la loi. Il peut également ordonner sur requête toutes mesures urgentes lorsque les circonstances exigent qu'elles ne soient pas prises contradictoirement. Les requêtes afférentes à une instance en cours sont présentées au Président de la chambre à laquelle l'affaire a été distribuée ou au juge déjà saisi.

N'ayant aucune formation de juriste, je quitte ce terrain miné de l'acquisition de données dans un cadre légal. Je ne souhaite pas donner de mauvaises interprétations de ces articles de loi. Avocats, juristes, huissiers de justice, etc., peuvent vous renseigner sur les difficultés juridiques et les aspects pratiques de ces saisies. Retournons à la technique.

Hors sujet : Remarque, dans les cas d'endettement, un ordinateur est considéré comme bien insaisissable s'il sert à l'activité professionnelle. Sinon il peut être saisi pour être revendu (saisie-vente), mais les étapes de ces procédures donnent le temps au saisi de réaliser une copie de ses données **[SVENTE]**.

Se préparer à copier des données

Faire attention à ne pas aggraver un problème matériel

Une alimentation électrique faiblarde/en fin de vie peut être à l'origine d'une panne d'un disque dur. Si vous suspectez ce problème ou bien un problème physique lié à la machine, utilisez un autre ordinateur pour copier le disque et récupérer vos données. Côté température, les disques durs récents supportent assez bien les températures un peu élevées, mais sont sensibles aux températures basses **[GOO2007]**.

SMART

Il est possible de contrôler et d'observer les systèmes de stockage en utilisant la technologie SMART (*Self-Monitoring, Analysis and Reporting Technology system*) intégrée dans les disques durs PATA, SATA, SCSI, ainsi que certains Raid matériels. Dans le but d'anticiper les pannes, la technologie SMART surveille et informe de l'état de certains indicateurs de fiabilité comme la température, le nombre de secteurs réalloués, les erreurs de localisation des secteurs, le temps cumulé d'utilisation, etc. Remarque, les disques connectés en USB ou en *firewire* ne peuvent pas être surveillés dans la majorité des cas.

Quelques conseils :

- Activer la surveillance SMART dans la configuration (*setup*) du BIOS. Quand l'option SMART est activée, si une opération d'écriture échoue, les secteurs défectueux sont désactivés et les données sont immédiatement écrites sur des secteurs de secours.
- Utiliser un logiciel pour surveiller les informations SMART afin de détecter les problèmes physiques :
  - Smartmontools **[SMARTMON]** – logiciel *opensource* pour Windows, Linux, FreeBSD, NetBSD, OpenBSD, Solaris, Darwin, OS/2.
  - SMARTReporter **[SMARTREP]** – logiciel opensource pour Apple Macintosh.
- Si le disque est connecté via USB ou firewire, les informations SMART peuvent ne pas être rapportées. Si c'est le cas, il est conseillé d'utiliser une connexion directe de type SATA, PATA, SCSI, etc.

Voici un exemple de *reporting* SMART pour un Raid matériel Compaq composé de disques Seagate, le disque est en bonne santé. Sa température est actuellement assez faible, 32°C, grâce au système de climatisation.

```
smartctl -a /dev/cciss/c0d0 -d cciss,0
smartctl version 5.36 [i686-redhat-linux-gnu] Copyright (C) 2002-6 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

Device: COMPAQ   BF07288285      Version: HPB2
Serial number: 3KP07MRF00007536XVTL
Device type: disk
Transport protocol: Parallel SCSI (SPI-4)
Local Time is: Thu Nov 15 20:09:55 2007 CET
Device supports SMART and is Enabled
Temperature Warning Enabled
SMART Health Status: OK

Current Drive Temperature:     32 C
Drive Trip Temperature:        68 C
Elements in grown defect list: 0
Vendor (Seagate) cache information
  Blocks sent to initiator = 451086139
  Blocks received from initiator = 128032659
  Blocks read from cache and sent to initiator = 1083141278
  Number of read and write commands whose size <= segment size = 822106214
  Number of read and write commands whose size > segment size = 0
Vendor (Seagate/Hitachi) factory information
  number of hours powered up = 20534.87
  number of minutes until next internal SMART test = 28

Error counter log:
           Errors Corrected by           Total   Correction     Gigabytes    Total
               ECC          rereads/    errors  algorithm      processed    uncorrected
           fast | delayed   rewrites  corrected  invocations   [10^9 bytes]  errors
read:          0        0         0         0          0          0.000           0
write:         0        0         0         0          0          0.000           0

Non-medium error count:       23

SMART Self-test log
Num  Test              Status                 segment  LifeTime  LBA_first_err [SK ASC ASQ]
     Description                              number   (hours)
# 1  Background short  Completed                  -       0               - [-   -    -]

Long (extended) Self Test duration: 1064 seconds [17.7 minutes]
```

Voici le monitoring d'un disque SATA de la machine qui me sert à écrire cet article. Ici, côté température, on constate que le disque a atteint les 55°C (appart sous les toits un jour d'été...). Heureusement, il semblerait que les disques soient beaucoup plus résistants aux températures élevées.

```
smartctl -A /dev/sda
smartctl version 5.37 [i686-redhat-linux-gnu] Copyright (C) 2002-6 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

=== START OF READ SMART DATA SECTION ===
SMART Attributes Data Structure revision number: 10
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED
RAW_VALUE
  1 Raw_Read_Error_Rate     0x000f   057   054   006    Pre-fail  Always       -
212994225
  3 Spin_Up_Time            0x0003   096   096   000    Pre-fail  Always       -       0
  4 Start_Stop_Count        0x0032   100   100   020    Old_age   Always       -       1
  5 Reallocated_Sector_Ct   0x0033   100   100   036    Pre-fail  Always       -       0
  7 Seek_Error_Rate         0x000f   085   060   030    Pre-fail  Always   -   337285762
  9 Power_On_Hours          0x0032   093   093   000    Old_age   Always   -   6454
 10 Spin_Retry_Count        0x0013   100   100   097    Pre-fail  Always   -   0
 12 Power_Cycle_Count       0x0032   099   099   020    Old_age   Always   -   1482
194 Temperature_Celsius     0x0022   048   055   000    Old_age   Always   -   48
195 Hardware_ECC_Recovered  0x001a   057   053   000    Old_age   Always   -   212994225
197 Current_Pending_Sector  0x0012   100   100   000    Old_age   Always   -   0
198 Offline_Uncorrectable   0x0010   100   100   000    Old_age   Offline  -   0
199 UDMA_CRC_Error_Count    0x003e   200   200   000    Old_age   Always   -   0
200 Multi_Zone_Error_Rate   0x0000   100   253   000    Old_age   Offline  -   0
202 TA_Increase_Count       0x0032   100   253   000    Old_age   Always   -   0
```

La colonne `RAW_VALUE` contient une donnée brute spécifique à chaque constructeur, voire à des modèles. Il faut donc lire les colonnes normalisées `VALUE`, `WORST`, `THRESH`, respectivement la valeur courante, la moins bonne valeur mesurée et le seuil d'alerte lorsqu'il y en a un.

Prenons l'exemple de l'attribut `Reallocated_Sector_Ct`. La table `P-list`, *permanent-list* ou *primary-list*, contient la liste des secteurs défectueux détectés lors du processus de fabrication du disque, ces secteurs sont ignorés par l'électronique du disque, leur présence n'a pas d'impact sur les performances du disque. Il nous reste donc les secteurs défectueux qui apparaissent selon les conditions d'utilisation du disque. La technologie SMART place les coordonnées des secteurs devenus défectueux dans la table `G-list`, *growth-list*. Tout accès à l'adresse d'un de ces secteurs redirige automatiquement l'opération de lecture/écriture vers un secteur sain de secours, cette opération se ressent au niveau des performances, mais, autrement, le disque continue de fonctionner normalement. Cependant, la table `G-list` a une capacité limitée, l'attribut `Reallocated_Sector_Ct` indique qu'il reste 100 secteurs de secours, lorsque cette valeur sera descendue à 36 ou moins, le BIOS ou tout utilitaire de monitoring SMART devra émettre une alerte demandant un changement immédiat du disque.

Faire attention à ne pas modifier les systèmes de fichiers

Le simple fait de lister le contenu d'un répertoire ou de lire le contenu d'un fichier modifie les dates d'accès. On risque aussi la création de fichiers temporaires, la création de miniatures pour les `images/pdf/...`, l'indexation du contenu. Le simple fait de brancher un média déclenche sur la plupart des ordinateurs la détection d'une table des partitions et le montage des systèmes de fichiers (affectation d'une lettre d'unité sous Windows, montage en lecture/écriture dans `/media` sous Unix). Que ce soit une `FAT16/FAT32/NTFS/ext2/ext3/...`, chacun de ces systèmes de fichiers est modifié dès qu'il est rendu accessible en lecture/écriture : un flag est positionné pour indiquer que le volume est « *dirty* ». Ainsi, si le média est déconnecté ou l'ordinateur éteint incorrectement, ce drapeau sera détecté automatiquement lorsque le média sera connecté/l'ordinateur redémarré. Les systèmes de fichiers d'origine Unix `ext2/ext3/jfs/reiserfs/xfs/...` comportent de plus un compteur s'incrémentant à chaque fois que le volume est monté en lecture/écriture.

Préserver l'intégrité

La majorité des clés USB et certains disques durs ont un cavalier pour mettre le média en lecture seul. C'est une précaution facile à mettre en place pour éviter une écriture accidentelle sur le disque.

Sous Linux, la commande `hdparm` propose de mettre en lecture seule un périphérique :

```
[root@test ~]# hdparm -r 1 /dev/sdb

/dev/sdb:
 setting readonly to 1 (on)
 readonly     = 1 (on)
```

Cependant, cela n'a eu aucun effet en pratique lors de mes tests...

Dans un cadre forensique, il est préférable de copier le média vers un fichier image. Ainsi, aucun accès en écriture ne sera effectué par le système d'exploitation.

Préserver le contexte

La configuration de l'ordinateur dont le disque est copié est souvent importante : date/heure/fuseau horaire et écart avec une source fiable sont nécessaires si on souhaite établir une chronologie ; la disposition, l'ordre des disques et le modèle de contrôleur pour reconstituer un RAID, le type de connexions (par exemple, PATA *primary master*) pour identifier correctement les points de montage ou les lettres d'unités.

Quoi copier ?

La nature des informations à copier dépend des informations à récupérer ou à réparer.

- pour récupérer un fichier endommagé, il faut une copie de ce fichier, voire du système de fichiers ; des copies de sauvegardes du fichier, même si le fichier a été modifié depuis, peuvent aider à reconstituer le fichier ;
- pour récupérer un fichier effacé, il faut une image du système de fichiers ou au minimum de l'espace libre de celui-ci ;
- pour un système de fichiers endommagé ou reformaté, une image de celui-ci ;
- pour une partition supprimée, une image de l'espace non alloué ou une image du disque (physique ou logique) qui contenait cette partition ;
- pour une table des partitions corrompue ou vide, une image du disque (physique ou logique) ;
- pour un volume logique (LV) effacé, une image du volume group auquel il appartenait ;
- pour un volume group (VG) effacé, une image des volumes physiques qu'il contenait ;
- pour un disque logique (RAID), une image des disques physiques, leurs ordres, la configuration du Raid, le type de contrôleur Raid ou bien ses caractéristiques (*offset* des données, taille des blocs, système de parité) ;
- pour un disque comportant des secteurs défectueux, une image complète des partitions impactées ou du disque selon l'étendu des dégâts.

Cependant en pratique, c'est parfois encore plus complexe. Si on prend comme exemple la commande `chkdsk` de Windows, elle ne fonctionne que sur des systèmes de fichiers identifiés par une lettre d'unité. Certains outils commerciaux sont même artificiellement bridés pour ne faire que de la récupération de fichiers sur des cartes mémoires, d'autres que sur les CD/DVD et, enfin, certains que sur les disques durs. Il faudra donc au final parfois recopier la sauvegarde sur un support analogue au média original.

Zone constructeur des disques ATA (SATA/PATA)

Un système d'exploitation utilise la commande bas-niveau `IDENTIFY DEVICE` pour connaître la taille d'un disque. Cependant, si une zone constructeur `Host Protected Area` (HPA) existe à la fin du disque, cette taille est inférieure à la taille réelle fournie par la commande `READ NATIVE MAX ADDRESS`. Cette limite est paramétrable avec la commande `SET MAX ADDRESS`. La zone HPA a été introduite avec le standard ATA-4 **[ATA-4]** en 1998.

Cette zone HPA est utilisable pour y placer divers utilitaires comme ceux assurant la restauration disque à l'état initial. Cela a d'autres applications comme le système antivol lojack `[LOJACK]`. Il place une petite partie de son code au niveau du BIOS (*Zone Core Managed Environment* (CME) sur les BIOS Phœnix) qui se charge d'appeler le reste du programme stocké dans la zone HPA du BIOS de l'ordinateur. Ainsi, même si l'ordinateur est volé et le disque reformaté, il suffit qu'au moment du *boot* l'ordinateur soit connecté à un réseau pour que le programme se connecte à un site de l'éditeur, l'adresse IP source donne alors le moyen de localiser l'ordinateur. Cette zone peut aussi avoir des utilisations illégitimes, comme servir à dissimuler des données illégales ou y cacher un *rootkit*.

Dans un cadre forensique, il est donc important de réaliser une copie complète du disque, y compris de cette zone. `disk_stat` du SleuthKit **[SLKIT]** détecte la présence de cette zone et `disk_sreset` (amélioration de setmax **[SETMAX]** sous Linux) peut la supprimer temporairement, mais ils ne gèrent que les disques de moins de 130 Go. La norme ATA-6 **[ATA-6]** introduit le support LBA 48 pour gérer les disques de plus de 130 Go, de nouvelles commandes `READ NATIVE MAX ADDRESS EXT` et `SET MAX ADDRESS EXT` sont à utiliser, malheureusement la dernière version 2.0.9 du SleuthKit ne les gère pas. Il faut donc utiliser des outils constructeurs (**[HITACHI]**, **[SEAGATE]**) pour manipuler cette HPA sur les disques de grandes capacités.

Autre nouveauté de la norme ATA-6, elle introduit le *Device Configuration Overlay* (DCO). Il permet au constructeur de spécifier la taille du disque telle qu'elle sera retournée par la fonction `READ NATIVE MAX ADDRESS`. Ainsi, le constructeur peut proposer des disques aux caractéristiques identiques au secteur prêt. Là encore, il y a moyen de dissimuler des informations par ce moyen.

Cas particulier des clés USB

Par rapport à un disque dur, il n'y a aucun élément mobile dans une clé USB. Elle est donc bien plus résistante aux chocs et à la poussière. Coté système d'exploitation, une clé USB est vue de manière similaire à un disque dur. Un contrôleur intégré à la clé USB se charge des accès à la mémoire flash. La mémoire est

découpée en pages de 512 ou 2048 octets. Chaque page peut être lue individuellement. Pour l'écriture, c'est un peu plus complexe : chaque écriture est précédée par une phase d'effacement et ces deux opérations s'effectuent uniquement sur des blocs de pages, généralement des blocs de 32 pages de 512 octets ou 64 pages de 2048 octets. De technologie NAND, chaque bloc de mémoire supporte un maximum d'un million de cycles d'effacement/réécriture avant de devenir corrompu, le contrôleur gère donc intelligemment l'organisation des données sur la mémoire flash pour maximiser la durée de vie de la mémoire en évitant que certains blocs subissent beaucoup plus de cycles d'effacement/réécriture que d'autres. Des données occupant des secteurs consécutifs ne seront donc pas forcément contiguës au niveau de cette mémoire. En cas de « secteurs » défectueux, il est inutile de réitérer les tentatives de lecture, les conditions de lecture seront exactement les mêmes. Il n'y a pas d'élément mécanique ; il n'y a donc pas de temps perdu significatif à chaque erreur de lecture.

Attention, cela ne s'applique pas aux disques branchés par USB.

Comment réaliser des copies ?

Attention, si vous copiez l'intégralité d'un disque, la destination doit aussi être un disque et non une partition. Si vous copiez une partition, la partition destination n'a pas besoin d'être formatée. En revanche, elle doit être suffisamment grande. Si la destination est plus grande que la source, il est conseillé d'initialiser la destination avec des zéros. De fait, copier l'intégralité du disque sur un disque vierge est fortement recommandé.

dd : une méthode classique

`dd` est une commande standard sous Unix de copie de données brutes, utilisable pour la copie de tout fichier et disque (« Sous Unix, tout est fichier »). Il existe aussi une version pour Windows **[DDWIN]**. La syntaxe est « assez » simple : `dd if=source of=destination`. Pour initialiser la destination avec des zéros : `dd if=/dev/zero of=/dev/dev_destination`. De façon à ce que la copie ne s'arrête pas si la source comporte des secteurs défectueux, il faut rajouter une option, la syntaxe devient : `dd if=source of=destination conv=noerror`.

Exemple de copie :

- d'un disque vers un disque :

```
dd if=/dev/sda of=/dev/sdb conv=noerror ;
```

- d'une partition vers une partition :

```
dd if=/dev/sda1 of=/dev/sdb1 conv=noerror ;
```

- d'un disque vers un fichier image

```
dd if=/dev/old_disk of=image_disque.dd conv=noerror.
```

Pour accélérer le processus de copie, au lieu de lire et écrire secteur par secteur, vous pouvez ajouter `bs=8k`. Cela permettra de lire/écrire sur le disque dur par bloc de 8 k. Cependant, cela présente un inconvénient : il suffit d'un seul secteur défectueux pour que la lecture d'un bloc de données échoue. Dans ce cas, `dd` écrit un bloc de zéros et passe à la suite, ignorant la possibilité que des secteurs sains aient pu être présents.

dd_rescue

`dd_rescue` **[DDRESCUE]** pallie ce problème de la commande `dd` : si la lecture d'un bloc de données échoue, `dd_rescue` effectue une lecture secteur par secteur. Ainsi, cette commande bénéficie de la rapidité liée à la lecture de blocs de grandes tailles sans sacrifier son efficacité. La syntaxe est `dd_rescue source destination`. `dd_rescue` peut enregistrer dans un fichier la liste des erreurs de lecture, ainsi que la liste des secteurs défectueux. Utilisé conjointement avec le script `dd_rhelp`, une récupération de ces secteurs défectueux est retentée.

`dd_rescue` permet aussi de générer des fichiers creux (*sparse*) : `dd_rescue -a source destination`. Si un bloc ne comporte que des zéros, `dd_rescue` choisit alors de ne pas réaliser d'écriture pour ce bloc. Ainsi, ce bloc est non alloué au niveau du système de fichiers, et donc le fichier destination occupe moins de place qu'un fichier classique. Les blocs non alloués sont lus comme comportant uniquement des zéros. Le fichier se comporte de manière identique à un fichier classique. Attention, il ne faut jamais utiliser cette option si le fichier destination existe déjà, sans quoi la source et la destination peuvent ne pas être identiques à la fin de la copie.

Une meilleure méthode : GNU ddrescue

Dans un premier temps, `GNU ddrescue` **[GNUDDRESC]** copie les données en utilisant une taille de bloc assez importante pour offrir une bonne rapidité. Ensuite, il lit les zones endommagées secteur par secteur, et, enfin, réessaye plusieurs fois de lire les secteurs défectueux. `ddrescue` récupère donc les données plus rapidement. Notons, par rapport à `dd_rescue`, qu'il ne crée pas de fichier creux. La syntaxe est `ddrescue source destination ddrescue.log`.

ewfacquire

Des produits commerciaux comme Forensic Toolkit (FTK) **[FTKIMG]** ou EnCase Forensic **[ENCASE]** ont leurs propres formats d'acquisition de données. Ces formats ajoutent des sommes de contrôle pour vérifier l'intégrité des données, des métadatas pour documenter l'acquisition, ainsi que de la compression pour que l'image prenne moins de place. J'utilise ici `ewfacquire` pour réaliser une image d'un DVD. Cet utilitaire vient avec `libewf` **[LIBEWF]**, une bibliothèque open source pour gérer l'*Expert Witness Compression Format* (EWF). La commande `ewfacquire` réalise donc une image d'un média suivant de nombreux paramètres contrôlant divers aspects, tels que la compression et la gestion des erreurs de lecture.

```
./ewfacquire /dev/hda
ewfacquire 20070512 (libewf 20070512, zlib 1.2.3, libcrypto 0.9.8, libuuid)

Acquiry parameters required, please provide the necessary input
...
The following acquiry parameters were provided:
Image path and filename:        test_dvd.E01
Case number:                    1
Description:                    Copie d'un DVD
Evidence number:                1
Examiner name:                  Christophe Grenier
```

```
Notes:
Media type:                     removable
Volume type:                    physical
Compression used:               none
Compress empty blocks:          no
EWF file format:                EnCase 5
Acquiry start offet:            0
Amount of bytes to acquire:     290390016
Evidence segment file size:     665600 kbytes
Block size:                     64 sectors
Error granularity:              64 sectors
Retries on read error:          2
Wipe sectors on read error:     yes

Continue acquiry with these values (yes, no) [yes]:

Acquiry started at: Mon Nov 12 21:59:42 2007

This could take a while.

Status: at 0%.
        acquired 32 kB (32768 bytes) of total 276 MB (290390016 bytes).

Status: at 1%.
        acquired 2.8 MB (2916352 bytes) of total 276 MB (290390016 bytes).
        completion in 6 minute(s) and 36 second(s) with 708 kB/s (725975
        bytes/second).
...
Acquiry completed at: Mon Nov 12 22:00:22 2007

Written: 276 MB (290390016 bytes) in 40 second(s) with 6.9 MB/s (7259750 bytes/
         second).

MD5 hash calculated over data:  62c5d9a293af1401d821fcefc5fb4b6d
```

Afflib

En réponse à ces formats propriétaires, *Advanced Forensic Format* (AFF) **[AFFLIB]**, un format ouvert a été créé. Il permet d'ajouter des métadatas personnalisées et de générer plusieurs fichiers plutôt qu'un seul fichier énorme (EnCase en est aussi capable). La compression LZMA apporte des gains de place de l'ordre de 30% par rapport à un fichier EnCase, mais, par défaut, la compression gzip est utilisée.

```
Elapsed Time: 00:00:11              IMAGING            Mon Nov 12 22:22:24 2007
Source device:  /dev/hda              AFF Output: test_dvd.aff
Model #:
                                        Disk Size:  290 MB (1024 byte sectors)
                                    Total sectors: 283,584




[==========>
  ]

 Currently reading sector:          32,768 (32768 sector chunks)  (11.55% done)

          blank sectors:                0
               Done in:          00:01:28 (this drive)


            Bytes read:       33,554,432
         Bytes written:       26,028,978
```

```
Overall compression ratio:          22.43%  (0% is none; 100% is perfect)

Free space on capture drive: 2,959 MB
                                    WRITING   ===>


./aimage --no_preview /dev/hda test_dvd.aff
****************************** IMAGING REPORT ******************************
Input: /dev/hda

  AFF Output file: test_dvd.aff
  Bytes read: 290,390,016
  Bytes written: 259,361,289

raw image md5:  62C5 D9A2 93AF 1401 D821 FCEF C5FB 4B6D
raw image sha1: FA82 4799 2EB8 79C2 B5FA 7BC2 0FC7 3A06 C935 2D69
raw image sha256: 19EF B16B D7AB 919E DC40 B28E 4772 151D 9081 C6A8 B1BC B475 5D84 514A 3E6D 29F4
Free space remaining on capture drive:  2,737 MB
```

Attention

Par défaut, les adresses MAC des cartes réseau ainsi que la sortie de dmesg (journal système) de la machine servant à l'acquisition figurent dans les métadatas.

e2image

Certains outils sont spécialisés pour ne copier que les métadatas. Les données ne sont pas copiées, uniquement la structure du système de fichier : informations sur l'arborescence (noms des fichiers, répertoire, taille), sur l'allocation des données. `e2image` **[E2FSPROGS]** réalise ceci pour les systèmes de fichier ext2/ext3. Une copie des métadatas est suffisante pour réaliser une liste des évènements ayant eu lieu : accès/modification/création/suppression de fichiers. C'est parfois un compromis acceptable si la copie complète du disque est impossible.

ntfsclone

`ntfsclone` **[NTFSPROGS]** est en premier lieu un outil de sauvegarde permettant de réaliser une copie rapide d'une partition NTFS vers un format spécifique ou un fichier creux (sparse), mais aussi un outil de duplication rapide, comme son nom l'indique. Comme `e2image` pour les systèmes ext2/ext3, seules les métadatas sont copiées.

Résumé sur les outils

Voir Tableau 1, page suivante.

Vérifier l'intégrité

Il est possible d'utiliser des sommes de contrôle `md5`, `sha1`, `sha512` pour calculer une empreinte d'un fichier ou d'un média. Voici un exemple sous Linux avec des commandes standards (*core-utils*)

Utilitaire	Avantages	Inconvénients
dd	En standard sous Linux/MacOSX/*BSD/	Mauvaise gestion des secteurs défectueux
dd_rescue	Gère les secteurs défectueux	Nécessite le script dd_rhelp
GNU ddrescue	Gère les secteurs défectueux efficacement	
ewfacquire	Forensique, supporte les formats EnCase	Les formats EnCase ne sont pas supportés par le SleuthKit
afflib	Forensique, introduit un format libre	Supporté par le SleuthKit
e2image	Duplication rapide des systèmes de fichiers ext2/ext3	N'est utilisable en forensique que pour établir une liste d'événements
ntfsclone	Duplication rapide des systèmes de fichiers NTFS	N'est utilisable en forensique que pour établir une liste d'événements

TI

```
[kmaster@ads1 1-extend-part]$ md5sum ext-part-test-2.dd
ea19519fc310835dd1fc20fc3cf36481  ext-part-test-2.dd
[kmaster@ads1 1-extend-part]$ sha1sum ext-part-test-2.dd
859e2e8a7dbd98d336469dc06ae9b3ca627e7b49  ext-part-test-2.dd
[kmaster@ads1 1-extend-part]$ sha512sum ext-part-test-2.dd
3b19ef5b733cf7a415319baa10c67dee271b2bb5af637ce45a38fd6ac873557c1b4a5c86d81aee6fe3ba11839a8dc5
683c8940d038d6ac8792d43b2fd2a18495  ext-part-test-2.dd
```

Ces empreintes numériques permettent de détecter toute modification accidentelle. Si l'empreinte est identique pour le média d'origine et sa copie, on peut considérer que la copie est identique à l'originale. Il est calculatoirement difficile de modifier un fichier tout en gardant la même empreinte numérique. Cependant, l'utilisation des algorithmes `md5` et même `sha1` n'est plus conseillée au vu de l'avancée des récentes attaques cryptographiques.

Accéder aux fichiers d'une image

Sur un système de type Unix, `mount -o loop,ro image.dd /mnt/image` rend accessibles les fichiers présents sur l'image du système de fichiers. L'option `ro` (`ro`=*read-only*) restreint l'accès à un accès en lecture seule. Ainsi, on ne risque pas de modifier cette image.

Conclusion

Cet article vous a présenté quelques outils du monde libre permettant de dupliquer des supports, endommagés ou non, aussi bien dans le but d'une récupération de données que dans une optique légale. Ne négligez pas cette étape, même si cela prend du temps et monopolise des ressources. Cela peut être le seul moyen de récupérer vos données ou d'obtenir une saisie valide.

Références

[AFFLIB] *AFF: The Advanced Forensic Format*, **http://www.afflib.org/**

[ATA-4] *AT Attachment with Packet Interface Extension (ATA/ATAPI-4)*, **http://www.t10.org/t13/project/d1153r18-ATA-ATAPI-4.pdf**

[ATA-6] *AT Attachment with Packet Interface Extension (ATA/ATAPI-6)*, **http://www.t10.org/t13/project/d1410r3a-ATA-ATAPI-6.pdf**

[DDRESCUE] dd_rescue, utilitaire de copie de disque par Kurt GARLOFF, **http://www.garloff.de/kurt/linux/ddrescue/**

[DDWIN] DD for Windows, **http://www.chrysocome.net/dd**

[DUPUY2007] Indiscrétions et « zones constructeurs » des disques durs, **http://actes.sstic.org/SSTIC07/Indiscretions_Zones_Constructeurs_Disques_Durs/SSTIC07-Dupuy-Indiscretions_Zones_Constructeurs_Disques_Durs.pdf**

[E2FSPROGS] e2fsprogs

[ENCASE] EnCase Forensic, **http://www.guidancesoftware.com/products/ef_index.asp**

[FTKIMG] FTK Imager, **http://www.accessdata.com**

[GNUDDRESC] GNU ddrescue, utilitaire de copie de disque par Antonio DIAZ, **http://savannah.gnu.org/projects/ddrescue/**

[GOO2007] *Failure Trends in a Large Disk Drive Population*, **http://labs.google.com/papers/disk_failures.pdf**

[HITACHI] *Feature Tool, modification zone* HPA, **http://www.hitachigst.com/hdd/support/download.htm**

[LIBEWF] **https://www.uitwisselplatform.nl/projects/libewf/**

[LOJACK] Lojack, système antivol pour portable, **http://www.lojackforlaptops.com/**

[NTFSPROGS] Ntfsprogs, ensemble d'utilitaires pour gérer des volumes NTFS, **http://www.linux-ntfs.org/doku.php?id=ntfsprogs**

[SLKIT] SleuthKit, **http://www.sleuthkit.org/**

[SMARTMON] *Smartmontools, monitoring SMART*, **http://smartmontools.sourceforge.net/**

[SMARTREP] *SMARTReporter, monitoring SMART*, **http://homepage.mac.com/julianmayer/**

[SVENTE] Saisie-Vente, **http://www.ump.assemblee-nationale.fr/article_texte.php3?id_article=1660**

[SETMAX] *Setmax, modification zone* HPA, **http://www.win.tue.nl/~aeb/linux/setmax.c**

[SEAGATE] *DiscWizard Starter Edition, modification zone* HPA, **http://www.seagate.com/www/en-us/support/downloads/discwizard**

Analyse post mortem tout en mémoire sous windows

Forensique mémoire sous Windows

Nicolas RUFF
nicolas.ruff@eads.net
Samuel DRALET
s.dralet@lexfo.fr

Cet article tente d'aborder la problématique de la forensique mémoire en environnement Windows. Avant même que l'analyse forensiques n'ait débuté, l'expert se retrouve face à des problématiques techniques engendrées par les différentes versions de Windows non compatibles entre elles. Quelles sont les solutions techniques à disposition de l'expert ? Peut-il se contenter des outils disponibles sur Internet ? Quelles sont les limites d'une analyse forensiques tout en mémoire ? Ce sont à toutes ces questions que nous essaierons de répondre.

mots clés : ***forensique / autopsie / mémoire / Windows / post-intrusion***

Objectifs de la forensique mémoire

Disque vs Mémoire

Les techniques de la forensique « traditionnelle » (orienté disque) sont bien connues et maîtrisées. De nombreux produits matériels (ex. : copieurs de disques) et logiciels sont disponibles à la vente. Et ces techniques produisent des résultats tangibles, reconnus devant les tribunaux.

Dans ces conditions, on peut se demander à quoi sert la forensique mémoire, une technique encore jeune et immature, dont les résultats sont difficiles à exploiter (surtout dans un environnement propriétaire comme Microsoft Windows) et contestables (le support étant souvent altéré pendant la collecte).

De notre point de vue, la forensique mémoire adresse des problématiques complètement différentes de la forensique « légale » : il s'agit, lors d'une réponse à un incident de sécurité, de collecter les traces les plus volatiles de l'intrusion, afin d'identifier la méthode d'intrusion de l'attaquant, et les dommages causés au système.

On parle ici bien souvent d'intrusions ciblées, dans lesquelles l'attaquant cherche à être le plus discret possible (contrairement aux intrusions « classiques », où l'attaquant cherche à compromettre le plus grand nombre de sites avec la même faille – cf. compromission récente de plusieurs milliers de sites européens en un week-end [1]).

L'intrusion passe la seconde

Pour un attaquant cherchant à minimiser son empreinte sur le système, plusieurs outils sont disponibles « sur étagère » : Meterpreter [2] s'il est désargenté, Immunity Canvas ou Core Impact s'il est sponsorisé par une institution. Grâce à ces outils, il lui est possible d'effectuer son forfait sans jamais écrire sur le disque, de manipuler les journaux système ou de changer la date d'accès aux fichiers (sur laquelle se basent la plupart des outils « disque » pour établir la *timeline* de l'intrusion).

De récents travaux (présentés lors de la conférence *Black Hat* 2007 : [3][4]) s'intéressent également aux intrusions complexes dans les bases de données, et à leur autopsie. Leur conclusion est la même : compte tenu de la complexité des formats de fichiers sous-jacents, un outil d'analyse disque (qui n'est jamais qu'une interface graphique pour `grep`, en caricaturant) ne trouvera rien. Il est nécessaire de sauvegarder le contexte mémoire du serveur de base de données pour obtenir l'information la plus « fraîche ».

Méthodes d'investigation

Contexte d'intervention

Il n'existe pas de méthode unique permettant de collecter à coup sûr une image fiable de la mémoire sur un système en cours d'exécution, tout en minimisant l'impact sur le système autopsié. Plusieurs facteurs entrent en jeu dans la détermination du processus de collecte.

1▶ Quelle est la configuration matérielle de la machine ?

Selon la possibilité d'accéder physiquement à la machine compromise et la connectique disponible, il est possible d'envisager une extraction de données matérielle (via le bus FireWire par exemple). Si aucune connectique « intéressante » (= autorisant les accès DMA) n'est disponible, il faudra s'orienter vers une solution 100% logicielle.

2▶ Quelle est la version du système Windows ?

Selon la version de Windows installée, les API disponibles ne sont pas les mêmes. Le cas le plus connu est celui du périphérique spécial `\Device\PhysicalMemory` ou de l'appel système `ZwSystemDebugControl`, ces deux vecteurs ayant été bloqués en mode utilisateur à partir de Windows 2003 SP1 (et subséquemment Windows XP SP2) – voir à ce sujet [5].

Sur les systèmes antérieurs, un outil tel que « dd » [6] permet de *dumper* la mémoire physique à partir du périphérique spécial. Sur les systèmes plus récents, le chargement d'un *driver* est nécessaire.

Le cas extrême est celui des versions « 64 bits » de Windows, sur lesquelles tous les accès à la mémoire physique en mode utilisateur sont bloqués, et où les drivers doivent être signés pour être chargés dans un système en configuration nominale (dite « compatible contenu *premium* » ;).

3▶ Quelle est la configuration mémoire utilisée ?

De nombreux paramètres affectent la manière dont Windows gère la mémoire physique disponible.

Si la mémoire est collectée via un `crashdump` (avec *dump* mémoire complet), il est nécessaire que le fichier d'échange (`pagefile`) ait une taille au moins égale à la mémoire physique disponible, sachant que la collecte de la mémoire au-delà du 4ème gigaoctet nécessite de toute façon un amorçage du système avec l'option `/MAXMEM` **[7]**.

Si la fonction d'hibernation (`suspend to disk`) a été préalablement activée, cette méthode permet de collecter la mémoire allouée tout en limitant l'impact opérationnel sur le serveur (redémarrage possible sans crash).

Enfin, lors de la phase d'analyse, connaître la valeur des paramètres suivants du fichier `BOOT.INI` est critique pour une reconstruction de l'espace d'adressage virtuel :

- `/PAE` (*Physical Address Extension*, nécessaire sur les systèmes disposant de plus de 4 Go de RAM ou souhaitant exploiter le bit « NX »[1] des processeurs récents).
- `/3GB` permettant de réserver plus d'espace utilisateur aux applications.

Des options subtiles du Memory Manager[2], telles que `DisablePagingExecutive`, vont également impacter la facilité de reconstruction de la mémoire.

4▶ La machine a-t-elle été préalablement préparée pour une collecte ?

Rentrent dans cette catégorie les systèmes de collecte matérielle (par exemple sous forme de carte PCI, bien que celles-ci restent marginales actuellement), et les options de configuration facilitant le travail de l'intervenant (telles que la taille du fichier d'échange vue précédemment ou l'option `/EMS` du fichier `BOOT.INI`).

5▶ L'activité de la machine peut-elle être interrompue ?

C'est souvent un point chaud, en particulier sur les serveurs de base de données. Provoquer le crash du système pour en collecter la mémoire peut laisser les données dans un état instable (et aucune sauvegarde n'est disponible, bien entendu). Dans certains cas, l'interruption de service n'est pas envisageable même après une intrusion.

6▶ L'agent de collecte peut-il installer des drivers ?

Cette question peut paraître stupide (sauf sur les systèmes 64 bits). Néanmoins, sur le terrain, il arrive que personne n'ait le mot de passe « administrateur » sur le serveur (ex. : externalisation des services informatiques) ou qu'une configuration locale et/ou des outils de sécurité interdisent l'installation de nouveaux drivers…

Méthodes de collecte

Les avantages et les inconvénients des méthodes de collecte ont été présentés en détail lors de la conférence *SSTIC* 2007, dont les actes sont en ligne **[8]**.

La technique la plus couramment utilisée consiste à crasher le système et à générer un dump mémoire complet. Afin d'obtenir le maximum d'informations, il est nécessaire au préalable de collecter le fichier d'échange. À notre connaissance, aucun outil gratuit ne le permet, mais les travaux d'IvanLef0u sont un bon point de départ pour un développement personnel **[9]**.

Parmi les techniques exotiques, on peut citer la fonction d'hibernation (mais, seules les pages mémoire allouées sont disponibles) **[10]**. On peut également citer l'injection en mémoire d'un nouveau système d'exploitation complet **[11]**, en l'occurrence un coLinux **[12]**, permettant d'effectuer la collecte en parallèle du système Windows. Intellectuellement intéressante, cette technique reste pour l'instant au stade du laboratoire.

Analyse

Là encore, les principales techniques d'analyse ont été présentées lors de *SSTIC* 2007 **[8]**. Elles se classent en deux catégories :

- L'analyse avec l'outil Microsoft WinDbg. Cet outil connaît toutes les structures internes du noyau et dispose de *plugins* puissants. Néanmoins, il n'est pas orienté forensique et ne permet donc pas facilement de reconstruire les binaires à partir des processus, en particulier lorsque ceux-ci sont terminés.
- L'analyse avec des outils tiers, qui permettent une analyse plus poussée des zones mémoire libérées, mais moins précise, car toutes les structures de données de toutes les versions de Windows n'ont pas encore été reconstruites par la communauté du *reverse engineering*.

En pratique

Lire la mémoire

Qu'on veuille dumper la mémoire ou faire de la forensique mémoire en live sur un système en cours d'exécution, la principale (et l'unique ?) problématique de la forensique mémoire finalement est la suivante : comment lire cette « foutue » mémoire sous Windows. Prenant en considération le contexte d'intervention et les différentes contraintes que cela implique, on s'aperçoit que peu de solutions techniques sont facilement utilisables dans la réalité.

Même si la technique la plus couramment utilisée consiste à crasher le système et à générer un dump mémoire complet, il n'est pas toujours possible de procéder de la sorte sur des serveurs en production. Quelles solutions reste-t-il en pratique ?

1. La solution VMWare

C'est la solution optimale, celle que tout le monde souhaite. Il suffit de mettre la machine virtuelle en pause (ce qui est tout à fait envisageable sur des serveurs en production) et de récupérer le fichier `.vmem`. Qui peut espérer meilleur scénario ? Et avec la virtualisation qui se démocratise de plus en plus, cette solution a l'avantage d'être de plus en plus courante.

[1] *NoExecute : pages mémoire non exécutables*
[2] *Clés HKLM\System\CurrentControlSet\Control\Session Manager\Memory Management**
[3] *Emergency Management Services*

2. La solution DD

La seconde solution est d'avoir le périphérique *PhysicalMemory* accessible en mode utilisateur et de pouvoir dumper la mémoire à l'aide de l'outil DD [6]. Mais, avant même de commencer l'analyse, on rencontre un premier « problème ». Il est stipulé sur le site de l'auteur que la nouvelle version de DD ne supporte pas PhysicalMemory :

Remark 11: The version of DD distributed with this release does not support \\.\PhysicalMemory pseudo-device as input.

Il est donc nécessaire de se munir de l'ancienne version de DD et la procédure devient finalement simple :

```
C:\dd.exe if=\\.\PhysicalMemory of=D:\file.dd conv=noerror --md5sum
--verifymd5 --md5out=D:\file.dd.md5
```

3. L'API NtSystemDebugControl()

Cette API permet, comme son nom l'indique, de réaliser des opérations de débogage directement au niveau du noyau. Parmi ses différents arguments, il y a `ControlCode` qui permet de spécifier ce que l'on souhaite faire. Si on donne comme valeur à `ControlCode` : `DebugSysReadVirtual`, on peut alors lire la mémoire du système (par déduction, `DebugSysWriteVirtual` pour y écrire).

IvanLef0u a décortiqué cette API et fournit le détail de son analyse et un exemple de code source sur son blog [13]. Il existe également l'outil memimager.exe [14] disponible sur Internet.

Malheureusement, comme cela a été dit précédemment, l'utilisation de cette API n'est plus autorisée depuis Windows 2003 SP1. Certains outils comme Nigilant32 de la société Agile Risk Management [15] affirment pouvoir dumper la mémoire sur Windows 2003, mais ne précisent pas si c'est avec ou sans SP1. Un rapide test et le message d'erreur « *Unable to open Physical Memory* » nous renseignent très vite sur les limites de cet outil.

Finalement, pour pouvoir lire la mémoire sur Windows 2003 SP1, la meilleure solution reste d'implémenter la technique d'Alex Ionescu [5] (vulnérabilité dans la Virtual DOS Machine ou VDM), malheureusement complexe et susceptible de disparaître dans les futures versions de Windows.

4. L'utilisation d'un driver

System Virginity Verifier 2.3 de Joanna Rutkowska fournit le code source d'un driver dans lequel est implémentée une fonction pour lire la mémoire en mode noyau (`readVirtualMem()` dans `vmemory.cpp`). Même si cette solution s'avère efficace, elle présente néanmoins des défauts.

D'une part, il est toujours dangereux de fonctionner en mode noyau et nous ne sommes pas à l'abri de faire crasher le système. Sur un système en production, il est évident que nous n'avons pas droit à l'erreur. Comme le décrit Joanna dans sa présentation de BH Federal 2006 [16], la fonction `MmProbeAndLockPages()` peut entraîner une `ACCESS_VIOLATION` et `MmIsAddressValid()` introduit une condition temporelle. De plus, cette dernière API ne permet pas d'accéder aux pages *swappées* (bien qu'il soit tout à fait possible d'implémenter sa propre fonction `MmIsAddressValid()`).

D'autre part, en installant un driver sur le système, on ne respecte pas la première règle de toute analyse forensique, à savoir ne rien installer sur le système autopsié pour garantir une intégrité maximale des données.

Analyser la mémoire

En pratique et dans un contexte d'analyse forensique réelle, très peu de solutions s'offrent à nous, surtout pour les systèmes Windows 2003 SP1 (sans parler de Windows Vista qu'il reste à étudier).

Supposons malgré tout que la mémoire du système à analyser puisse être lue. Deux scénarios sont alors possibles : analyser la mémoire à partir d'une image ou directement sur le système en cours d'exécution.

La première solution peut être un choix judicieux, puisque la plupart des outils d'analyses forensiques disponibles sur Internet travaillent sur une image mémoire. Seulement, si vous souhaitez effectuer une analyse à un instant t, t+1, t+2, etc., il est alors nécessaire de dumper à chaque fois la mémoire. L'analyse forensique peut vite devenir fastidieuse.

La seconde solution, à savoir analyser le système en cours d'exécution, peut rendre l'analyse plus efficace en permettant d'acquérir davantage de preuves informatiques. Il est en revanche nécessaire soit de modifier les outils existants, soit de réimplémenter toutes les techniques d'analyse connues afin de pouvoir agir sur un système actif... de quoi faire peur.

Exemple d'analyse mémoire

Prenons l'exemple de l'outil PTFinder [8] d'Andreas Schuster. Cet outil va entre autres extraire les structures `_EPROCESS` d'un dump mémoire. Il permet ainsi de lister les processus actifs, mais aussi les processus terminés et cachés. Le mode opératoire est le suivant :

1▶ Il recherche la chaîne de caractères `03001b00` dans le dump mémoire. `0x03` est la valeur du champ `Type` de la structure `_DISPATCHER_HEADER` et `0x1b` du champ `Size`. Cette structure est présente dans la structure `_EPROCESS` définissant un processus :

```
lkd> dt _EPROCESS -r
   +0x000 Pcb              : _KPROCESS
      +0x000 Header           : _DISPATCHER_HEADER
         +0x000 Type             : UChar
         +0x001 Absolute         : UChar
         +0x002 Size             : UChar
         +0x003 Inserted         : UChar
         +0x004 SignalState      : Int4B
         +0x008 WaitListHead     : _LIST_ENTRY
[...]
```

D'après Andreas [19], ces valeurs sont fixes depuis Windows 2000 jusqu'à Windows 2003 inclus. Pour Vista, le champ `Size` change et est égal à 0x20.

2▶ À chaque occurrence de la chaîne `03001b00`, il effectue certaines vérifications pour éliminer un maximum de faux positifs, notamment :

- Le `PageDirectoryBase` (ou PDB) équivalent à la valeur du registre CR3 est utilisé pour convertir les adresses virtuelles en adresses physiques. Ce champ ne doit pas être nul.
- Le PDB doit être aligné sur une page de 4 ko.
- La valeur du champ `ThreadListHead.Flink` qui pointe vers une structure doit être supérieure à 0x80000000, c'est-à-dire être dans l'espace noyau. Le champ `ThreadListHead` contient une structure `List Entry` qui lie tous les *threads* d'un processus.
- La valeur du champ `ThreadListHead.Blink` doit aussi être supérieure à 0x80000000.

3. À partir de l'*offset* de la chaîne `03001b00`, il est alors possible de récupérer toutes les informations d'un processus utiles à une analyse forensique (PID, nom du binaire, etc.), en connaissant les différents offsets des champs de la structure `_EPROCESS`. Le souci est que cette structure est différente en fonction de la version du système Windows. Andreas (toujours lui) a documenté dans plusieurs billets sur son blog **[20]**, les structures `_EPROCESS` (et `_ETHREAD`) en fonction des versions des Windows.

4. En récupérant les différentes structures `_EPROCESS` d'un dump mémoire, il est aussi possible de connaître les processus terminés. Ils se caractérisent par un champ `ExitTimeLo` et `ExitTimeHi` non nul. Quant aux processus cachés, ils ne le sont plus du fait que PTFinder ne parcourt pas la liste doublement chaînée des structures `_EPROCESS`.

Malgré quelques faux positifs, le concept de PTFinder semble assez efficace pour détecter les techniques de dissimulation connues, comme la technique DKOM ou encore le *rootkit* Futo. Malheureusement, PTFinder nécessite une image mémoire et ne peut agir sur des systèmes en live. Nous avons donc développé un outil qui utilise l'API `NtSystemDebugControl()` pour lire la mémoire et la technique de PTFinder pour rechercher des structures `_EPROCESS`.

```
C:\> memiris.exe

[...]
% help eprocess

eprocess [ -ltsecgdop ] <arguments>

This command makes it possible to display _EPROCESS structures
via kernel memory.

__options
  -o [os] : choose os between { xpsp2, xp, w2k, 2003 }
  -l : list process
  -t : list terminated process
  -s [addr] : scan from start address
  -e [addr] : scan to end address
  -d [addr] : display _EPROCESS structure
  -p [pid] : display pid's PEB
  -c : check _EPROCESS structures
  -g : for /3GB systems
```

Quelques exemples d'utilisation

À titre d'exemple, nous cachons le processus `calc.exe` grâce à la technique DKOM (*Direct Kernel Object Manipulation*) sur un système Windows 2000 à l'aide de l'outil d'IvanLef0u **[13]**.

```
C:\> kfist.exe calc.exe
[...]
Process Name : calc.exe
Process Found ! Now hiding it...
Prev EPROCESS : 0x811df020
Next EPROCESS : 0x815fb2f0
Hiding done, have fun!
EPROCESS : 0x815fb2f0
```

On liste les processus actifs sur le système :

```
% eprocess -o xp -l -s 0x80000000 -e 0x90000000 -c

Addr                 Pid                  Binary
------------------------------------------------------------
0x80551d80             0                    Idle
0x811df020           324                 cmd.exe
0x813006c8          1976              dumprep.exe
0x81395730          1964                PSPad.exe
0x813d79b8           440               logonui.exe
0x814208d8          1024                logon.scr
0x8144bda0          1380            IEXPLORE.EXE
0x8145f928           624              memiris.exe
0x814926d0          1324                calc.exe
[...]
```

On affiche la structure `_EPROCESS` à l'adresse `0x814926d0` du processus `calc.exe` précédemment caché :

```
% eprocess -o xpsp2 -d 0x814926d0

[#] Structure _EPROCESS at 0x814926d0
Pid: 1324
Ppid: 1736
Binary: calc.exe
PageDirectoryBase: 0x071682e0
PEB: 0x7ffde000
ThreaListHead.Blink : 0x81630d98
ThreadListHead.Flink : 0x81630d98
ActiveProcessLinks.Blink: 0x811df0a8
ActiveProcessLinks.Flink: 0x815fb378
CreateTime: Fri Nov 23 17:12:10 2007
ExitTime: (null)
```

On affiche la structure PEB du processus `calc.exe` :

```
% eprocess -p 1324

[#] PEB process pid 1324
BeingDebugged: No
Image Base : 0x01000000
WindowTitle: C:\WINDOWS\system32\calc.exe
ImageFile: C:\WINDOWS\system32\calc.exe
CommandLine: "C:\WINDOWS\system32\calc.exe"
```

```
DllPath: C:\WINDOWS\system32;C:\WINDOWS\system32;C:\WINDOWS\system;C:\
        WINDOWS;.;
C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\Program Files\
                  Microsoft Platform SDK for Windows XP SP2\Bin\.;C:\Program
                  Files\Microsoft Platform SDK for Windows XP SP2\Bin\
                  WinNT\.;C:\Program Files\Microsoft Platform SDK for Wind
                  ows XP SP2\Bin\.;C:\Program Files\Microsoft Platform SDK
                  for Windows XP SP2\Bin\WinNT\.

Module          Base Addr     Entry Pt      Size   Load Count      Type
-------------------------------------------------------------------------
calc.exe        0x01000000    0x01012475    0x0001f000          -1     Static
ntdll.dll       0x7c910000    0x7c923156    0x000b7000          -1     Static
kernel32.dll    0x7c800000    0x7c80b5ae    0x00105000          -1     Static
SHELL32.dll     0x7c9d0000    0x7c9f7366    0x00823000          -1     Static
[...]
```

On stoppe le processus `calc.exe` et on affiche tous les processus terminés :

```
% eprocess -o xp -t

Addr             Pid       Binary                    Exit Time
-------------------------------------------------------------------
0x813086c8      1976    dumprep.exe     Fri Nov 23 11:02:21 2007
0x813d79b8       440    logonui.exe     Fri Nov 23 16:07:58 2007
0x814208d8      1024      logon.scr     Fri Nov 23 15:24:27 2007
0x8144bda0      1380   IEXPLORE.EXE     Fri Nov 23 11:02:17 2007
0x814926d0      1324       calc.exe     Fri Nov 23 17:23:09 2007
```

Conclusion

Comme cet article essaie de le démontrer, la collecte et l'analyse de la mémoire sur un système Windows relève du grand art, mais permet de détecter beaucoup plus finement des traces d'intrusion que ne le permet une analyse de disque.

Compte tenu de la recrudescence des attaques « ciblées », les techniques de la forensique mémoire ont connu un essor considérable ces dernières années.

Mais, dans ce domaine de la forensique plus que dans tout autre, la rapidité de réaction est un élément prépondérant. Inutile d'espérer des résultats probants sur un serveur compromis depuis 3 mois et rebooté plusieurs fois depuis... d'où l'intérêt d'un outil de *live forensic* exécuté régulièrement.

Références

[1] *Large scale European Web Attack*,

http://www.websense.com/securitylabs/alerts/alert.php?AlertID=782

[2] Metasploit's Meterpreter,

http://www.metasploit.com/projects/Framework/docs/meterpreter.pdf

[3] LITCHFIELD (David), « *Database Forensics* », *Black Hat* 2007

https://www.blackhat.com/presentations/bh-usa-07/Litchfield/Presentation/bh-usa-07-litchfield.pdf

[4] FOWLER (Kevvie), « *SQL Server Database Forensics* », *Black Hat* 2007

https://www.blackhat.com/presentations/bh-usa-07/Fowler/Presentation/bh-usa-07-fowler.pdf

[5] IONESCU (Alex), « *Subverting Windows 2003 SP1 Kernel Integrity Protection* », *RECON* 2006

http://recon.cx/en/f/aionescu-subverting-w2k3-kernel-integrity-protection.ppt

[6] *Forensics Acquisition Utilities*,

http://www.gmgsystemsinc.com/fau/

[7] « *How to overcome the 4,095 MB paging file size limit in Windows,* »

http://support.microsoft.com/kb/237740

[8] RUFF (Nicolas), « Forensics mémoire sous Windows », *SSTIC* 2007

http://actes.sstic.org/SSTIC07/Forensics_Memoire_Windows/SSTIC07-Ruff-Forensics_Memoire_Windows.pdf

[9] Ivan Lef0u, « *Pagefile Attack* »,

http://ivanlef0u.free.fr/?p=77

[10] RUFF (Nicolas), SUICHE (Matthieu), « Enter Sandman », *PacSec* 2007

http://www.pacsec.jp/

[11] SCHATZ (Bradley), « *BodySnatcher: Towards reliable volatile memory acquisition by software* », *DFRWS* 2007,

http://dfrws.org/2007/proceedings/p126-schatz_pres.pdf

[12] Cooperative Linux,

http://www.colinux.org/

[13] Ivan Lef0u, « *NtSystemDebugControl Demystified* »,

http://www.ivanlef0u.tuxfamily.org/?p=21

[14] **http://www.ntsecurity.nu/toolbox/memimager/**

[15] Agile Risk Management : Nigilant32,

http://www.agilerm.net/publications_4.html

[16] RUTKOWSKA (Joanna), « *System Virginity Verifier* »,

http://invisiblethings.org/tools.html

[17] Volatile Systems : *The Volatility Framework*,

http://www.volatilesystems.com/VolatileWeb/volatility.gsp

[18] SCHUSTER (Andreas), « PTFinder »,

http://computer.forensikblog.de/en/2006/09/ptfinder_0_3_00.html

[19] SCHUSTER (Andreas), « _DISPATCH_HEADER »,

http://computer.forensikblog.de/en/2006/02/distpatcher_header.html

[20] SCHUSTER (Andreas), « *More on processes and threads,* »

http://computer.forensikblog.de/en/2006/02/more_on_processes_and_threads.html

5/5

Retour d'expériences

Retour d'expériences d'analyses forensiques

La réussite ou l'échec d'une analyse forensique dépend fortement de l'expérience que l'on a dans ce domaine. Il y a tellement d'éléments à prendre en considération que personne n'est vraiment à l'abri d'une erreur. Zythom, un expert judiciaire, témoigne très clairement à ce sujet sur son blog [6]. Il doit intervenir sur les ordinateurs d'une entreprise ayant mis la clé sous la porte. Il prépare comme il se doit la mission, mais oublie malgré son expérience une chose primordiale... vérifier qu'il y a bien de l'électricité dans l'entreprise.

Pensez à tous les cas de figures qui peuvent se présenter lors d'une analyse forensique est une des difficultés parmi tant d'autres. L'article tentera, par un retour d'expériences, de montrer les problèmes techniques et juridiques qu'il

mots clés : ***forensique / contraintes techniques / contraintes juridiques***

Classification des analyses forensiques

D'après les différentes missions effectuées, on distingue deux catégories d'analyse *forensique* :

- L'analyse est spécifique au support informatique et à la problématique du client. La récupération d'*event logs* effacés sur un Windows 2000 (le support informatique est donc ici un disque dur) et la récupération de SMS effacés sur une carte SIM sont typiquement deux exemples d'analyses forensiques complètement distincts de par le support informatique à analyser et la problématique à résoudre.
- L'analyse est « générique », puisqu'une seule problématique existe : prouver qu'il y a intrusion sur le système d'information du client concerné. Quel que soit ce client, l'analyse forensique est toujours méthodologiquement identique et s'effectue toujours avec les mêmes outils : analyse de logs, audit de l'activité sur le système piraté, détection de *rootkit* et *backdoor*, etc.

Cette classification n'a évidemment rien d'officiel. C'est uniquement un constat personnel par rapport aux problématiques des différentes analyses forensiques réalisées. De cette classification dépendent les outils à utiliser, les procédures à mettre en œuvre et les problèmes rencontrés lors des analyses forensiques. Ces problèmes peuvent être d'ordre technique, mais aussi juridiques selon que l'analyse forensique ait été effectuée en tant qu'expertise judiciaire ou en tant que prestation informatique.

Différents cas concrets d'analyses forensiques vont être détaillés. Elles ont toutes été réalisées sous forme de prestations informatiques. Elles serviront à expliquer plus clairement la classification précédemment détaillée et les problèmes techniques et juridiques rencontrés.

Les analyses spécifiques

Récupération d'event logs effacés

Le contexte est le suivant : une tierce personne a volontairement détruit les event logs d'un serveur Windows 2000 en effaçant chaque enregistrement à l'aide de l'observateur d'évènements. Les différents fichiers `.evt` sont toujours présents, seuls leurs contenus ont été effacés. Le client nous ayant prévenu quelques jours après l'incident, il est encore fort probable de pouvoir récupérer des informations pertinentes.

L'objectif de l'analyse est de récupérer les event logs et de prouver que cette personne s'est connectée un dimanche entre 14h et 17h. Bien entendu, le serveur est en production et aucun arrêt de service n'est toléré.

La technique pour récupérer des event logs effacés a déjà été expliquée dans un précédent article de MISC [1]. En résumé, la technique consiste à faire une copie bits à bits du disque dur et à analyser la copie à la recherche d'une chaîne de caractères propre au format des event logs de manière à les reconstruire.

Quels sont les éléments nécessaires à la préparation de cette analyse ? L'information primordiale à connaître en priorité est la capacité du disque dur ou de la partition primaire. Dans ce cas précis, le disque dur ne faisait **que** 60 Go. Il n'y avait donc aucun problème pour trouver un disque dur externe d'une capacité supérieure.

Maintenant, imaginez le contraire, une capacité de plusieurs centaines de gigaoctets de données à collecter. Comment effectuer une copie du disque dur ? Via le réseau ? Oui, mais il faut avoir sous la main un serveur avec suffisamment d'espace disque pour accepter la copie ? Si le client n'en possède pas, va-t-il accepter la connexion de votre ordinateur sur son réseau ? Je ne parle même pas des cas d'analyses forensiques avec des grappes de disques durs en Raid 5.

Il faut savoir ensuite comment effectuer le transfert des données d'un disque à l'autre. Le serveur étant en production, il est évident qu'on ne va pas démonter le disque dur (d'où l'utilisation d'un disque dur externe). Plusieurs solutions sont alors possibles, la plus intéressante étant de passer par un port USB facilement accessible physiquement. Ce fut le cas pour cette analyse. Quelles auraient été les solutions dans le cas contraire ? Transfert via un port série, via le réseau sur un autre serveur sur lequel est branché le disque dur externe, etc. ?

Vient ensuite la copie du disque dur. Une certaine pression s'installe, puisque le serveur est en production. Aucune erreur n'est tolérée. Il est conseillé d'avoir préparé les outils nécessaires, de les avoir tester, d'avoir établi une procédure exacte d'utilisation afin que tout se déroule pour le mieux.

Samuel Dralet
s.dralet@lexfo.fr

Reste ensuite l'analyse à proprement parler pour récupérer les event logs effacés. La connaissance de leurs formats est forcément nécessaire, sans quoi l'analyse peut échouer. Malgré tout, aucun outil de récupération ne répondait à nos besoins : la reconstruction des event logs fut longue et fastidieuse. Pour un disque dur de 60 Go, il a fallu environ 10h pour arriver à nos fins. Heureusement, nous connaissions les informations à rechercher et le client n'avait heureusement pas fixé de *deadline*.

Finalement, notre prestation a permis d'apporter l'expertise d'une société extérieure au dossier (c'était le souhait du client) et d'appuyer les différentes preuves déjà existantes. Notre intervention s'est arrêtée là.

Récupération de preuves informatiques via la technique dite de « file carving »

Le contexte est différent. Un employé est soupçonné d'envoyer des lettres injurieuses en interne. Les lettres contiennent des mots spécifiques et une photo modifiée qui vont servir à l'analyse. L'incident est vieux de **4 mois**. L'objectif de la prestation est d'apporter la preuve en analysant le poste de travail de l'employé qu'il est bien l'auteur de ces lettres.

Le souci de cette mission est l'absence de l'employé concerné lors de notre intervention. Avons-nous le droit légalement d'intervenir sur son poste de travail ? La réponse est « oui », sous certaines conditions. Il est nécessaire de préciser qu'en tant que prestataire nous nous engageons à exécuter la mission en respectant l'ensemble des dispositions légales et réglementaires en vigueur relatives à la protection des données personnelles et que le client est tenu de respecter ces mêmes règles vis-à-vis de ses employés. Nous devons conseiller le client dans ce sens. Nous sommes uniquement là pour répondre à une problématique technique en ayant pris soin d'avertir et de conseiller le client sur ces aspects juridiques.

Beaucoup d'informations au sujet de la protection des données à caractère personnel et de cybersurveillance sont disponibles sur le site de l'AFCDP [2].

Il reste maintenant la partie technique de la mission. Elle consiste à faire une copie du disque dur (avec les mêmes contraintes que l'analyse précédente) et à l'analyser de deux manières différentes :

- Recherche de chaînes de caractères dans le but de trouver les mots utilisés par l'employé.

  Cela peut sembler simple au premier abord, mais plusieurs difficultés sont présentes. Tout d'abord, il s'avère d'après l'équipe informatique du client que l'employé n'a pas sauvegardé la lettre sur son poste de travail. Il s'est contenté de taper la lettre et de l'imprimer. Une analyse au niveau des imprimantes pouvait être effectuée, mais nous étions autorisés à intervenir uniquement sur le poste de travail. Le second problème est venu du fait que le client nous a prévenus beaucoup trop tard. Il n'y avait quasiment aucune chance de retrouver des données temporaires en rapport avec cette lettre. Cette recherche s'est avérée finalement sans résultat et avec beaucoup de faux positifs.

- Recherche d'images selon la technique dite de « *file carving* » [3].

  L'outil utilisé lors de l'analyse est Scalpel [4] (il en existe évidemment d'autres comme Photorec). Il a permis de récupérer toutes les photos y comprises celles effacées. Et au lieu de retrouver la photo modifiée, c'est l'originale qui a été récupérée.

La photo originale est le seul élément récupéré, en rapport avec la lettre injurieuse et la photo modifiée, fourni au client. Maintenant rien ne prouve que la photo n'ait pas été copiée sur le poste de travail par une tierce personne, ou bien que ce fameux ordinateur n'ait pas été utilisé par une autre personne que la personne incriminée.

En résumé, nous avons joué notre rôle de conseiller :

- Nous avons effectué la mission sur un plan technique, à savoir rechercher des informations spécifiques sur un ordinateur.
- Nous avons mis en garde le client devant les risques juridiques.

Récupération de SMS effacés

La troisième et dernière analyse forensique est assez particulière dans son contexte. Une personne a récupéré le téléphone portable de son mari avec tous les SMS effacés. L'affaire est délicate, puisque cette personne souhaite prouver grâce à ces SMS que son mari a été poussé au suicide !!! L'objectif est de récupérer des SMS effacés sur une carte SIM (la méthode a déjà fait l'objet d'un article dans MISC [5], elle ne sera pas décrite de nouveau).

Les seules contraintes rencontrées ont été de nature technique. Nous décidons d'utiliser notre lecteur de carte SIM et l'outil Chipit pour récupérer d'éventuels SMS effacés. Malgré l'opération effectuée des dizaines de fois, il s'avère qu'il est impossible de lire la carte SIM :

- Chipit n'envoie pas les bonnes commandes à la carte SIM.
- Un autre outil plus « professionnel » (*Smart Access* de Atmel) est nécessaire pour envoyer ses propres commandes à la carte SIM.

La spécificité de la carte SIM nous a obligé à rechercher un autre outil et à étudier plus en détail le protocole pour lui envoyer des commandes et recevoir des données. La durée de l'analyse s'est avérée beaucoup plus longue que prévue. La carte SIM était une carte SFR. Malheureusement, sans une expérience sur ce type de carte, il est impossible d'éviter ce genre de problème.

Après lecture de tous les enregistrements de la carte SIM, uniquement les SMS visibles avec le terminal GSM ont pu être récupérés. Les autres enregistrements contenaient tous les octets FF. Pour pousser plus loin l'analyse, il aurait fallu analyser la mémoire du terminal GSM au risque de le détériorer. Le client ne l'a pas souhaité.

En résumé

La difficulté d'une analyse forensique dépend finalement :

- ⇨ du type du support informatique à analyser ;
- ⇨ de la taille des données à collecter ;
- ⇨ de la méthode pour collecter les données ;
- ⇨ des outils d'analyses disponibles ;
- ⇨ du contexte juridique.

Il suffit qu'un de ces éléments ne soit pas maîtrisé pour que l'analyse forensique échoue dans sa totalité.

Les analyses génériques : détection de compromission

La problématique

On parle d'analyses génériques pour les analyses dont la problématique reste identique quel que soit le client. Ce dernier vous appelle complètement alarmé : « j'ai des fichiers bizarres sur un de mes serveurs ! ». Le client sous-entend qu'il y a eu intrusion sur son système d'information. La problématique reste la même quel que soit le client : détecter toute forme d'intrusion potentielle. Il faut ensuite pouvoir répondre aux quatre questions suivantes : quand, comment, où et par qui a eu lieu l'intrusion. C'est ce que nous appelons la détection de compromission ou plus communément une réponse aux incidents de sécurité.

Les difficultés rencontrées

Sans parler des techniques de détection de *rootkits*, *backdoors*, etc., les analyses de type détection de compromission présentent néanmoins quelques difficultés auxquelles il faut faire face.

Une première difficulté réside dans le fait qu'il est nécessaire d'être préparé techniquement à la détection de compromission. En général, c'est un peu la panique quand un client détecte une intrusion. Il tente d'analyser lui-même ses serveurs à l'aide d'outils récupérés sur Internet et sans aucune méthodologie ou vous appelle pour effectuer l'analyse forensique.

Sans les outils et procédures adéquats, vous pouvez passer à côté de preuves d'intrusion ou même écraser ces preuves. Étant donné l'hétérogénéité des systèmes d'information des clients, les outils à utiliser ou à développer doivent être supportés sur un maximum d'architectures (Windows, Linux, Solaris, AIX en l'occurrence). Ensuite, ces outils doivent répondre à la règle élémentaire d'une analyse forensique : aucune modification du système.

Une seconde difficulté réside dans les systèmes à analyser. Du fait de l'hétérogénéité des systèmes d'information, il faut déjà s'adapter à chaque situation. Mais, le problème est l'évolution rapide des systèmes d'exploitation sans compatibilité descendante. Typiquement, se présente le cas du *device PhysicalMemory* accessible en tant qu'utilisateur uniquement sur certaines versions de Windows (l'article « Forensiques mémoire sous Windows » dans ce dossier en parle plus précisément). Vous avez aussi les différentes versions du noyau Linux. Entre les versions 2.4 et 2.6, les techniques d'analyses peuvent changer. Il est alors nécessaire de développer des outils spécifiques à chaque version.

Que dire des techniques d'intrusion qui évoluent sans cesse et qui laissent de moins en moins de traces sur un système ? Que dire aussi des analyses forensiques tout en mémoire et des difficultés que cela implique ? Le sujet est vaste et demanderait un article à lui tout seul tant les problèmes techniques d'analyses sont nombreux.

Un exemple d'analyse

Le client nous a demandé une analyse dans un but préventif : il souhaite vérifier que deux de ses serveurs sont vierges de toutes intrusions avant d'y appliquer une sécurité pour leur garantir une protection optimale. Ces deux serveurs hébergent chacun une application Web accessible depuis Internet.

Pour ne pas corrompre les deux serveurs à analyser et pour éviter un arrêt de service, la démarche suivante a été suivie afin de s'assurer du bon déroulement de l'analyse :

- ⇨ Copie (image VMWare) des deux serveurs : deux nouveaux serveurs sont créés.
- ⇨ Analyse forensique sur chaque copie des serveurs.
- ⇨ On s'assure que l'analyse forensique ne pose aucun problème sur les serveurs de recette.
- ⇨ Fermeture de la connexion Internet du premier serveur de production pour lancer l'analyse forensique. Puis, réouverture de la connexion Internet.
- ⇨ L'action est réitérée pour le second serveur de production.

Un arrêt de service est toujours le problème le plus redouté dans une analyse forensique. Procéder de cette manière en lançant l'analyse sur des serveurs de recette avant les serveurs de production afin s'assurer que les outils utilisés ne provoquent aucun crash des serveurs est une garantie non négligeable pour l'analyste. Il est d'ailleurs préférable d'effecteur des tests avec le même système d'exploitation que les serveurs du client avant de commencer l'analyse forensique sur ces derniers. Les risques sont minimisés. C'est typiquement une partie de la préparation de l'analyse forensique.

Mais, même en ayant préparé correctement cette analyse forensique, nous avons sous-évalué sa durée. Les outils, par exemple, collectent souvent beaucoup d'informations qu'ils doivent analyser et transférer via le réseau sur une tierce machine. Il est délicat d'estimer précisément la durée de l'analyse avant de l'avoir commencée. Dans ce cas précis, la bande passante du réseau, les CPU des serveurs ou encore leurs mémoires jouent un rôle non négligeable. On a beau faire tous les tests imaginables avant d'aller chez le client, il est impossible de reproduire à l'identique l'analyse forensique telle qu'elle se déroulera chez le client.

Finalement, il reste le problème du nombre impressionnant de systèmes d'exploitation différents qu'il est possible de rencontrer chez les clients (sont inclus les différentes versions d'un même système d'exploitation, par exemple Windows XP SP1 et Windows XP SP2). Cette hétérogénéité nous oblige à effectuer de longues et fastidieuses phases de tests des outils que nous utilisons, qu'ils soient disponibles sur Internet ou développés en interne.

Conclusion

Cet article a présenté un bref aperçu de difficultés rencontrées lors des analyses forensiques, fondé sur des expériences vécues. Mais, il est impossible d'en faire le tour tant les analyses forensiques sont spécifiques, que ce soient les investigations informatiques et les problématiques qu'elles doivent résoudre ou les détections de compromission et les systèmes d'exploitation à analyser.

Quelques conseils cependant :

- Obtenez un maximum d'informations techniques de la part du client avant de commencer l'analyse forensique.
- Préparez vos analyses forensiques, faites un maximum de tests avant l'analyse forensique, écrivez des procédures. Dans l'absolu, vous devez connaître chronologiquement les outils que vous allez utiliser avec leurs paramètres exacts.
- Développez si possible vos propres outils. Vous maîtriserez davantage vos analyses forensiques et vous n'irez pas au devant de mauvaises surprises avec des outils récupérés sur Internet.
- Et, finalement, assurez-vous que vos contrats sont « blindés » juridiquement.

Références

[1] DRALET (Samuel), « Les event logs », *MISC* 30,
http://www.lexfo.fr/blog/index.php/2007/03/22/5-article-sur-les-event-logs

[2] AFCDP,
http://www.afcdp.org

[3] Carving,
http://www.forensicswiki.org/wiki/Carving

[4] « *Scalpel: A frugal, high performance file carver* »,
http://www.digitalforensicssolutions.com/Scalpel/

[5] « Acquisition de données sur les mobiles », *MISC* 31
http://www.lexfo.fr/blog/index.php/2007/05/07/16-article-sur-l-acquisition-de-donnees-sur-les-mobiles

[6] Le siècle des lumières, Zythom
http://zythom.blogspot.com/2006/12/le-sicle-des-lumires.html

Comprenez les systèmes de sécurité d'OpenSolaris et Solaris

Minimisation des services réseau sur un poste de travail Solaris 9 et 10

Solaris 10 tout comme les versions qui l'ont précédé est disponible pour 2 architectures : Sparc et X86. Dans sa version X86/X64, il se place désormais comme une alternative envisageable comme poste de travail bureautique, puisqu'il opère sur un simple PC et offre les principaux outils exigés dans le cadre d'une telle activité : suite bureautique, navigateurs web accompagnés de quelques plugins, agents de messagerie électronique et de messagerie instantanée, etc. Dans ce cadre limité, il peut se mesurer à Windows, MacOS X et aux diverses distributions Linux avec un avantage de taille : il est à la fois gratuit et ouvert comme Linux tout en présentant un aspect industriel fort lié à son usage actuellement beaucoup plus répandu comme serveur.

mots clés : ***durcissement / système / configuration / sécurisation***

Jusqu'à l'*update* 2, Solaris 10 comme les versions qui l'ont précédé est installé avec un grand nombre de services réseau activés. À partir de Solaris 10 update 3, l'installation permet de choisir une configuration minimale adaptée aux postes de travail dans laquelle les services activés comme Xorg ou Sendmail sont sécurisés et où les services réseau activés sont limités. On pourra parfaire cette configuration en activant le `tcp_wrapper` intégré ou en configurant le coupe-feu IP Filter. Avant l'update 3 de Solaris 10, il est donc indispensable de désactiver les services non utilisés et de sécuriser ceux qui doivent être ouverts.

Solaris, qui n'avait pas évolué depuis l'origine (Solaris 2.0) sur le plan de la méthode de lancement des services réseau, introduit avec la version 10 de très nombreux changements dont SMF (*Service Management Facilities*) et sa commande `svcadm` qui permet d'activer/désactiver les services. L'update 3 met à profit cette commande et introduit la commande `netservices` qui sera utilisée pour passer d'un état fermé parfaitement adapté à un profil poste de travail (ou à un serveur sécurisé) à un état ouvert correspondant à une configuration de départ pour certains serveurs, et inversement. Un correctif rajoute cette fonctionnalité à tout système Solaris 10.

```
# netservices open
restarting syslogd
restarting sendmail
restarting wbem
dtlogin needs to be restarted. Restart now? [Y] y
restarting dtlogin

# netservices limited
restarting syslogd
restarting sendmail
restarting wbem
```

Avec une configuration fermée qui est l'état par défaut, seuls les services suivants sont ouverts :

```
# nmap rama
PORT    STATE SERVICE
22/tcp  open  ssh
111/tcp open  rpcbind
# rpcinfo -p rama
rpcinfo: can't contact portmapper: RPC: Authentication error; why = Failed
(unspecified error)

# nmap -sU rama
PORT      STATE         SERVICE
111/udp   open|filtered rpcbind
32771/udp open|filtered sometimes-rpc6
```

Le résultat avec une configuration ouverte est assez proche de celui obtenu avec une version 9 :

```
# nmap rama
...
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
79/tcp   open  finger
111/tcp  open  rpcbind
513/tcp  open  login
514/tcp  open  shell
587/tcp  open  submission
898/tcp  open  sun-manageconsole
4045/tcp open  lockd
6112/tcp open  dtspc
7100/tcp open  font-service

# nmap -sU rama
PORT      STATE         SERVICE
111/udp   open|filtered rpcbind
161/udp   open|filtered snmp
514/udp   open|filtered syslog
4045/udp  open|filtered lockd
32771/udp open|filtered sometimes-rpc6

# rpcinfo -p rama
   program vers proto   port  service
    100000    4   tcp    111  rpcbind
    100000    3   tcp    111  rpcbind
    100000    2   tcp    111  rpcbind
...
    100002    2   tcp  50959  rusersd
    100002    3   tcp  50959  rusersd
    100002    2   udp  39692  rusersd
    100002    3   udp  39692  rusersd
    300598    1   udp  39695
    300598    1   tcp  50962
```

Si Sun n'avait pas attendu l'update 3 de Solaris 10 pour prendre ces mesures, cet article pourrait presque se terminer ici. Néanmoins, maîtriser l'aspect configuration des services n'est pas inutile pour sécuriser les configurations plus anciennes et pour assurer un meilleur contrôle de la version courante.

Christian Pélissier

1. Autoriser ou interdire l'accès aux services réseau

1.1 Les services sous contrôle d'inetd

1.1.1 Cas de Solaris 9

La plupart des services placés sous le contrôle d'inetd sont des services non sécurisés. De ce fait, ils devront être désactivés ou soumis à un filtrage IP strict à l'exception du serveur de média éjectable utile pour l'éjection des CD-ROM. On s'assurera que le fichier `/etc/rmmount.conf` interdit la prise en compte des fichiers suid.

```
# /etc/rmmmount.conf
...
# Mount
mount * hsfs udfs ufs -o nosuid
```

Dans l'hypothèse où ce seul service est activé, le fichier `/etc/inet/inetd.conf` se réduira à la seule ligne :

```
# /etc/inet/inetd.conf
100155/1 tli rpc/ticotsord wait root /usr/lib/smedia/rpc.smserverd rpc.smserverd
```

Il est conseillé de conserver une copie du fichier `inetd.conf` original pour réactiver si nécessaire certains services.

1.1.2 Cas de Solaris 10

Sous Solaris 10, le démon inetd est un service particulier du système SMF. SMF remplace les procédures *shell* `/etc/rc[123].d/[KS]*` et traite inetd comme un service auxiliaire particulier. Les services SMF sont gérés à l'aide des commandes `svcs`, `svcadm`, `svccfg`, `svcprop`, etc.

Les services sous le contrôle d'inetd sont gérés à l'aide de la commande `inetadm`. Elle permet de lister les services réseau placés sous son contrôle, de modifier ses propriétés globales et les propriétés de chaque service. Elle répond à la syntaxe suivante :

```
# inetadm -?
Syntaxe :
  inetadm
  inetadm -?
  inetadm -p
  inetadm -l {FMRI | modèle}...
  inetadm -e {FMRI | modèle}...
  inetadm -d {FMRI | modèle}...
  inetadm -m {FMRI | modèle}... {name=valeur}...
  inetadm -M {name=valeur}...
```

Sans options, `inetadm` répertorie tous les services inetd placés sous son contrôle.

Options :

- `-?` : aide sur l'impression ;
- `-p` : répertorie toutes les valeurs de propriété inetd par défaut ;
- `-l` : répertorie toutes les valeurs de propriété inetd pour le(s) service(s) inet ;
- `-e` : active le(s) service(s) inet ;
- `-d` : désactive le(s) service(s) inet ;
- `-m` : modifie les valeurs de propriété inetd du ou des services inet ;
- `-M` : modifie les valeurs de propriété inetd par défaut.

Sur une station installée non sécurisée, on obtient la liste suivante :

```
# inetadm | grep online
enabled   online         svc:/application/x11/xfs:default
enabled   online         svc:/application/font/stfsloader:default
enabled   online         svc:/application/print/rfc1179:default
enabled   online         svc:/network/rpc/smserver:default
enabled   online         svc:/network/rpc/gss:default
enabled   online         svc:/network/rpc/mdcomm:default
enabled   online         svc:/network/rpc/meta:default
enabled   online         svc:/network/rpc/metamed:default
enabled   online         svc:/network/rpc/metamh:default
enabled   online         svc:/network/rpc/rstat:default
enabled   online         svc:/network/rpc/rusers:default
enabled   online         svc:/network/security/ktkt_warn:default
enabled   online         svc:/network/nfs/rquota:default
enabled   online         svc:/network/ftp:default
enabled   online         svc:/network/finger:default
enabled   online         svc:/network/login:rlogin
enabled   online         svc:/network/shell:default
enabled   online         svc:/network/rpc-100235_1/rpc_ticotsord:default
enabled   online         svc:/network/rpc-100083_1/rpc_tcp:default
enabled   online         svc:/network/rpc-100068_2-5/rpc_udp:default
```

Cette liste est plus étendue selon la mise à jour utilisée et les correctifs installés. Elle est à comparer à celle obtenue sur un équipement Solaris 10 U3 installé avec minimisation des services :

```
# inetadm | grep online
enabled   online         svc:/application/font/stfsloader:default
enabled   online         svc:/application/print/rfc1179:default
enabled   online         svc:/network/rpc/cde-calendar-manager:default
enabled   online         svc:/network/rpc/gss:default
enabled   online         svc:/network/rpc/smserver:default
enabled   online         svc:/network/security/ktkt_warn:default
enabled   online         svc:/network/rpc-100235_1/rpc_ticotsord:default
```

On voit qu'un effort a été fait, mais qu'il n'est pas complet, puisque au moins 2 services auraient pu être désactivés :

- `rfc1179` qui n'est utile que si l'équipement est serveur d'impression au travers du protocole LPD (rfc1179) ;

➪ `cde-calendar-manager` utile pour l'agenda CDE dtcm.

Que la machine soit installée sécurisée ou non, la désactivation des services indésirables se fera à l'aide d'une procédure qui une fois au point permettra de répéter l'opération.

```
#!/bin/ksh
inetadm -d svc:/application/x11/xfs
inetadm -d svc:/application/print/rfc1179
inetadm -d svc:/network/ftp
inetadm -d svc:/network/telnet
...
inetadm -d svc:/network/rpc/rex
inetadm -d svc:/network/rpc/rstat
inetadm -d svc:/network/rpc/rusers
inetadm -d svc:/network/rpc/spray
```

Dans le cas où aucun service inetd n'est utile, il est plus simple de désactiver inetd.

```
# svcadm disable svc:/network/inetd
```

1.2 Les services autonomes lancés par init

1.2.1 Cas de Solaris 9

Sous Solaris 9, le processus init lance les procédures commençant par la lettre S (comme `start`) situées dans les répertoires `/etc/rc[1S23].d`. Ces procédures sont en réalité des liens vers les fichiers du répertoire `/etc/init.d` qui les contient toutes.

La désactivation d'un service correspondant à une procédure donnée consiste à renommer le fichier en question. La seule contrainte est que le nom du nouveau fichier ne commence ni par la lettre S, ni par la lettre K. Si l'opération doit être répétée, on utilisera la procédure suivante qu'on adaptera :

```
#! /bin/ksh

cd /etc/rc2.d
for Svc in S73nfs.client S74autofs S90wbem
do
        /etc/rc2.d/$Svc stop
        mv $Svc NO_$Svc
done

cd /etc/rc3.d
for Svc in S76snmpdx S77dmi
do
        /etc/rc3.d/$Svc stop
        mv $Svc NO_$Svc
done
```

Il existe une autre méthode consistant à arrêter les services immédiatement après leur démarrage à l'aide d'un fichier `/etc/init.d/svcstop` dont le contenu sera par exemple le fichier ci-dessus (en supprimant les commandes `mv`) lié à `/etc/rc3.d/S99svcstop`. Elle présente l'inconvénient d'autoriser un bref démarrage des services et l'avantage d'être certain que le service sera toujours arrêté, car l'installation de patchs réinstalle régulièrement certains services comme S90wbem.

1.2.2 Cas de Solaris 10

Comme on l'a mentionné, Solaris 10 abandonne le système rc pour SMF. Les services SMF sont contrôlés à l'aide de la commande `svcadm`. Une autre commande `svcs` permet de lister l'état d'un ou de tous les services SMF, ainsi que ceux qui se trouvent sous le contrôle d'inetd tout comme ceux qui subsistent encore avec l'ancien système rc. On notera que ces derniers sont au fil des updates successifs intégrés dans SMF.

```
# svcs -a

...
legacy_run     Feb_04   lrc:/etc/rc3_d/S81volmgt
...
disabled       Feb_04   svc:/network/dns/client:default
...
online         15:54:02 svc:/network/smtp:sendmail
online         16:04:32 svc:/network/ipfilter:default
...
```

On active ou on désactive un service avec la commande `svcadm`. Le service peut être désigné par un nom plus court que le nom complet lorsqu'il n'y a pas ambiguïté comme :

```
# svcadm enable svc:/network/ipfilter:default
# svcadm enable ipfilter
# svcadm disable telnet
```

Dans le cas où l'on part d'une configuration ouverte, le script shell suivant permet de désactiver les services qui ne sont pas utiles sur un poste de travail et sur la plupart des serveurs. On l'adaptera aux diverses situations.

```
#!/bin/ksh

export PATH=/usr/sbin:/usr/bin:/usr/dt/bin

# SNMP et DMI

if [[ -f /etc/rc3.d/S76snmpdx ]]
then
        /etc/rc3.d/S76snmpdx stop
        mv /etc/rc3.d/S76snmpdx /etc/rc3.d/NO_S76snmpdx
else
        svcadm disable svc:/application/management/snmpdx:default
fi
if [[ -f /etc/rc3.d/S77dmi ]]
then
        /etc/rc3.d/S77dmi stop
        mv /etc/rc3.d/S77dmi /etc/rc3.d/NO_S77dmi
else
        svcadm disable svc:/application/management/dmi:default
fi

# DTLOGIN, XFS : selon l'update ou le niveau de correction de Solaris

dtconfig -kill; dtconfig -d

svcadm disable svc:/application/graphical-login/cde-login:default
svcadm disable svc:/application/x11/xfs

# LPD and IPP REMOTE PRINTING : si le poste n'est pas serveur d'impression
```

```
svcadm disable svc:/application/print/rfc1179
svcadm disable svc:/application/print/ipp-listener

# NFS Client : si le poste est client NFS commenter ces lignes

svcadm disable svc:/system/filesystem/autofs
svcadm disable svc:/network/nfs/status
svcadm disable svc:/network/nfs/nlockmgr
svcadm disable svc:/network/nfs/rquota
svcadm disable svc:/network/nfs/client

# Autres services agenda CDE et wbem

svc:/network/rpc/cde-calendar-manager:default
svc:/application/management/wbem:default
```

Une fois sécurisée, on listera les services ouverts :

```
# svcs -a | grep online

...
```

1.3 Journaliser et filtrer les services avec TCP wrapper

Les services concernés par le filtrage à l'aide du TCP wrapper sont aussi bien des services autonomes que ceux sous contrôle d'inetd.

1.3.1 Activation sous Solaris 9

Par défaut, la journalisation et le filtrage ne sont pas activés. L'activation sera effectuée au niveau du fichier `/etc/default/inetd`.

```
#
# /etc/default/inetd
#
ENABLE_CONNECTION_LOGGING=YES
ENABLE_TCPWRAPPERS=YES
```

1.3.2 Activation sous Solaris 10

L'activation sous Solaris 10 sera effectuée service par service et elle passe par l'utilisation de la commande `inetadm`. L'exemple suivant concerne l'activation de la journalisation et du filtrage pour le serveur telnet.

```
# inetadm -l telnet
SCOPE    NAME=VALUE
         name="telnet"
         endpoint_type="stream"
         proto="tcp6"
         isrpc=FALSE
         wait=FALSE
         exec="/usr/sbin/in.telnetd"
         user="root"
default  bind_addr=""
default  bind_fail_max=-1
default  bind_fail_interval=-1
default  max_con_rate=-1
default  max_copies=-1
default  con_rate_offline=-1
default  failrate_cnt=40
default  failrate_interval=60
default  inherit_env=TRUE
default  tcp_trace=FALSE
default  tcp_wrappers=FALSE
# inetadm -m telnet tcp_trace=TRUE tcp_wrappers=TRUE
# inetadm -l telnet
SCOPE    NAME=VALUE
         name="telnet"
...
default  inherit_env=TRUE
         tcp_trace=TRUE
         tcp_wrappers=TRUE
```

Cette opération sera à répéter avec tous les services inetd qu'on souhaite journaliser et filtrer. On remarquera qu'il est possible de lier avec le paramètre `bind_addr` un service à une adresse IP, ce qui est particulièrement intéressant si l'équipement est doté de plusieurs interfaces IP.

1.3.3 Filtrage sous Solaris 9 et 10

L'activation ne suffit pas, elle doit être complétée par la déclaration des services autorisés en renseignant les fichiers `/etc/hosts.allow` et `/etc/hosts.deny`. La configuration proposée permet de journaliser les succès dans syslog et d'envoyer une alerte par courrier électronique pour les échecs (attention toutefois, ce procédé n'est pas viable s'ils sont nombreux).

```
#
# /etc/hosts.allow
#

# %a  Adresse IP du client
# %h  Nom ou adresse IP du client
# %d  Nom du service
# %u  Nom utilisateur si connu, unknown sinon

sshd: 192.168.1.  : spawn ( /usr/bin/logger -t %d -p auth.notice -i "accepted connect
                    from %u@%h(%a)" )
in.ftpd: *.fr  : spawn ( /usr/bin/logger -t %d -p auth.notice -i "accepted connect from
                 %u@%h(%a)" )
#
# /etc/hosts.deny
#

ALL: ALL: spawn ( /usr/bin/mailx -s "refused connect on %s service from %u@%h(%a)"
          secadmin@localhost )
#in.tftpd: ALL: spawn ( /usr/bin/finger -l @%h | /usr/bin/mailx -s "%d-%h" root )
#ALL: ALL: spawn (spawn ( /usr/bin/logger -t %d -p auth.notice -i "refused connect from
           %h(%a)" ) &
```

1.4 Filtrer avec IPFilter

1.4.1 Solaris 9

IPfilter ne fait pas partie de la distribution de Solaris 9. Toutefois, les 2 modules qui le composent `pfil` et `ipfilter` se compilent très facilement et génèrent même 2 paquetages appelés `ipf` et `pfil` facilitant leur installation et leur maintenance. Une différence avec Solaris 10 réside dans le fait que les fichiers de configuration sont

situés dans /etc/opt/ipf avec Solaris 9 et dans /etc/ipf dans le cas de Solaris 10. Solaris 10 SMF contrôle le lancement de pfil et d'ipfilter, alors que Solaris 9 fait appel aux procédures /etc/rc2.d/S10pfil et /etc/rc2.d/S65ipfboot.

1.4.2 Solaris 9 et 10

Le logiciel IPFilter est un coupe-feu avec suivi d'état. Il est désactivé par défaut et il se configure à l'aide du fichier /etc/ipf/ipf.conf et /etc/ipf/pfil.ap. Pour améliorer la lisibilité des règles, on peut désigner des ensembles d'adresses avec /etc/ipf/ippool.conf.

Le fichier /etc/ipf/pfil.ap sert à déclarer les interfaces auxquelles seront associées un module de filtrage paquet filter. Ce fichier disparaît avec Solaris 10 update 4. Pour connaître le nom de l'interface, utilisez la commande netstat -i, puis activez la ou les interfaces listées en enlevant le commentaire approprié. Ce qui donnera par exemple :

```
#
# /etc/ipf/pfil.ap : IP Filter pfil autopush setup
#

#major  minor lastminor modules
#iprb   -1      0       pfil
#elxl   -1      0       pfil
e1000g  -1      0       pfil
#bge    -1      0       pfil
#nf     -1      0       pfil
...
```

À la suite de cette opération, on pourra lancer le module paquet filter et vérifier que le module pfil vient bien s'insérer entre la couche physique et la couche IP.

```
# svcadm enable svc:/network/pfil:default
# ifconfig e100g0 modlist
0 arp
1 ip
2 pfil
3 e1000g0
```

Le filtrage sera configuré à l'aide du fichier /etc/ipf/ipf.conf. Un filtrage de base correspond au fichier suivant :

```
#
# /etc/ipf/ipf.conf
#

# Tous les entrants sont bloqués (sans quick)

block in log level local0.notice on e1000g0 proto icmp all
block in log level local0.notice on e1000g0 proto tcp all
block in on e1000g0 proto udp all

# Tous les sortants sont bloqués (sans quick)

block out log level local0.notice on e1000g0 all

# Accès autorisés en entrée

# SSH on utilise un pool d'adresses défini dans ippool.conf

pass in log level local0.notice first quick on e1000g0 proto tcp from pool/10 to
any port = 22 flags S keep state

# HTTP/HTTPS
pass in quick on e1000g0 proto tcp from any to any port = 80 flags S keep state
pass in quick on e1000g0 proto tcp from any to any port = 443 flags S keep state

# NTP
pass in quick on e1000g0 proto igmp from any to 224.0.0.0/16
pass in quick on e1000g0 proto udp from any port = 123 to any port = 123 keep
state

# SMTP MTA et MSA
pass in quick on e1000g0 proto tcp from any to any port = 25 flags S keep state
pass in quick on e1000g0 proto tcp from any to any port = 587 flags S keep state

# Accès autorisés en sortie

# Clients TCP

pass out quick on e1000g0 proto tcp  from 192.168.1.3/32 to any keep state

# Clients UDP

pass out quick on e1000g0 proto udp  from 192.168.1.3/32 to any keep state
```

De son côté, le fichier ippool.conf sera utilisé pour déclarer des pools d'adresses IP.

```
#
# /etc/opt/ipf/ippool.conf
#

table role = ipf type = tree number = 10
{ 192.168.1.0/24, 192.168.16/32 };
```

L'activation du coupe-feu sera effective après cette commande :

```
# svcadm enable svc:/network/ipfilter
```

1.5 Configurer un serveur FTP anonyme sous Solaris 9 et 10

Il peut être utile d'ouvrir son poste de travail au dépôt de fichier. La solution la plus simple au niveau de sa gestion consiste à ouvrir un accès FTP anonyme. Avec ce type d'accès, l'utilisateur ne s'authentifie pas. Il n'accède par contre qu'à un environnement restreint par l'utilisation de l'appel système *chroot* qui lui donne une visibilité privée d'un sous-système de fichiers. Solaris offre depuis la version 9 la commande ftpconfig qui assure la création de l'environnement chroot. On notera qu'elle crée l'utilisateur ftp dans le groupe other.

```
# ftpconfig /export/home/ftp
Création de l'utilisateur ftp
Création du répertoire /export/home/ftp
Mise à jour du répertoire /export/home/ftp
# cd /export/home/ftp
```

```
# ls -l
total 10
lrwxrwxrwx   1 root     bin            9 avr  25 20:57 bin -> ./usr/bin
d--x--x--x   2 root     sys          512 avr  25 20:57 dev
d--x--x--x   5 root     sys          512 avr  25 20:57 etc
drwxr-xr-x   2 root     sys          512 avr  25 20:57 pub
d--x--x--x   5 root     sys          512 avr  25 20:57 usr
# mkdir .forward .rhosts; chmod 000 .forward .rhosts
# chown root:root .forward .rhosts
```

On pourra créer une zone de dépôt non listable pour assurer un minimum de confidentialité entre les déposants de la façon suivante :

```
# mkdir -p  private/nolist
# chmod 711 private
# chown root:root private
# chmod 733 private/nolist
# chgrp other private/nolist
# ls -ld private private/nolist
drwx--x--x   3 root     root         512 avr  25 21:08 private
drwx-wx-wx   2 root     other        512 avr  25 21:08 private/nolist
```

On notera que l'application de tout correctif concernant le système devra être suivi d'une mise à jour du FTP anonyme par :

```
# ftpconfig -d /export/home/ftp
```

Mise à jour du répertoire `/export/home/ftp`.

1.5.1 Le fichier /etc/ftpd/ftpusers

Ce fichier est la liste des utilisateurs auxquels on interdit l'accès FTP. Il faut donc l'initialiser avec la liste de tous les noms de *login*, sauf précisément `ftp`. On peut l'initialiser à l'aide de la commande suivante :

```
# cut -d":" -f1,1 /etc/passwd | grep -v "^ftp:" > /etc/ftpd/ftpusers
# chown root:root /etc/ftpd/ftpusers
# cat /etc/ftpd/ftpusers
root
daemon
bin
sys
...
```

1.5.2 Le fichier /etc/shells

L'accès FTP est autorisé si, pour l'utilisateur considéré, le shell de login existe dans `/etc/shells`.

```
/sbin/sh
/sbin/jsh
/bin/sh
/bin/jsh
/bin/ksh
/bin/csh
/usr/bin/sh
/usr/bin/jsh
/usr/bin/ksh
/usr/bin/csh
/usr/lib/rsh
/usr/bin/nsh
```

Pour un compte accessible seulement via FTP, on indiquera `/usr/bin/nsh` comme shell de login. On verra plus loin à quoi correspond ce shell.

1.5.3 Activation du FTP

Sous Solaris 10, si le service a été préalablement désactivé, on le réactivera par :

```
# svcadm enable ftp
```

Sous Solaris 9, ajoutez ou décommentez la ligne dans le fichier `/etc/inet/inetd.conf`.

```
ftp   stream   tcp6   nowait   root   /usr/sbin/in.ftpd   in.ftpd -a
```

La prise en compte sera effective après la commande `pkill -HUP inetd` et on n'oubliera pas de déclarer les adresses IP ou les domaines autorisés en ajoutant les informations appropriées dans le fichier `/etc/hosts.allow`.

1.6 Sécuriser Sendmail

Sur le poste de travail, les messages sont généralement lus en utilisant le protocole POP ou IMAP. Ils sont émis en utilisant le protocole SMTP, mais n'ont pas besoin d'être émis à partir du Sendmail local dès lors qu'on utilise un client évolué comme dtmail, Evolution ou Thunderbird. Toutefois, conserver Sendmail actif même si on ne l'utilise pas est indispensable pour la gestion des messages locaux comme ceux envoyés par les utilitaires `cron` et `at` ou pour remonter les alertes mailx générées par les règles du fichier `/etc/hosts.deny`.

1.6.1 Solaris 9

La solution la plus simple consiste à lier le démon Sendmail à l'adresse *loopback* 127.0.0.1. Cela peut être réalisé très simplement en générant un fichier `sendmail.cf` à partir du fichier M4 suivant :

```
divert(-1)dnl
# Configuration Sendmail sécurisée
divert(0)dnl
VERSIONID(`@(#)sendmail.mc')
OSTYPE(`solaris8')dnl
DOMAIN(`solaris-generic')dnl
define(`confFALLBACK_SMARTHOST', `mailhost$?m.$m$.')dnl
FEATURE(`no_default_msa')dnl
DAEMON_OPTIONS(`NAME=NoMTA4, Family=inet, Addr=127.0.0.1')dnl
DAEMON_OPTIONS(`Name=MSA4,   Family=inet, Addr=127.0.0.1, Port=587, M=E')dnl
MAILER(`local')dnl
MAILER(`smtp')dnl

LOCAL_NET_CONFIG
R$* < @ $* .$m. > $*    $#esmtp $@ $2.$m $: $1 < @ $2.$m. > $3
```

Le fichier généré viendra remplacer le fichier `/etc/mail/sendmail.cf`.

1.6.2 Solaris 10

Solaris 10 dans la configuration sécurisée opère avec un fichier `/etc/mail/local.cf` écoutant sur la seule adresse locale. La commande `ps -ef` permet de vérifier que l'on se trouve dans cette configuration.

1.7 Minimiser Apache

Le serveur Apache 1.3 n'est pas activé tant que le fichier de configuration `/etc/apache/httpd.conf` n'existe pas. Un gestionnaire de site Web peut avoir besoin d'un serveur Apache purement local lui permettant de mettre au point ses pages HTML et ses scripts CGI. Ici encore, il est possible de lier Apache 1.3 sur l'adresse 127.0.0.1. On pourra utiliser l'une de ces 2 directives.

```
Listen 127.0.0.1:80
Servername 127.0.0.1
```

Avec un serveur Apache 2.0.X (livré avec Solaris 10 uniquement), le démarrage du serveur est lié à la présence du fichier `/etc/apache2/httpd.conf`.

1.8 Interdire les accès distants au serveur X11

1.8.1 Solaris 9

Bien que les accès distants au serveur X11 contrôlables par échange de MAGIC-COOKIE ou à l'aide de la commande `xhost` soient fermés par défaut, il est possible de les désactiver complètement en supprimant la *socket* tcp et en conservant la socket locale indispensable au fonctionnement local.

```
# cp /usr/dt/config/Xservers /etc/dt/config
```

Dans le fichier `/etc/dt/config/Xservers`, on changera la déclaration d'origine commentée par celle qui suit :

```
#   :0  Local local_uid@console root /usr/openwin/bin/Xsun :0 -nobanner
:0 Local local_uid@console root /usr/openwin/bin/Xsun :0 -nobanner -nolisten tcp
```

Puis, on passera en mode terminal pour relancer le serveur dtlogin (XDM).

```
# /etc/rc2.d/S99dtlogin stop
# /etc/rc2.d/S99dtlogin start
```

1.8.2 Solaris 10

L'installation sécurisée de Solaris 10 U3 sécurise le démon Xorg en le lançant avec l'option `-nolisten tcp`.

1.9 Sécuriser le démon dtlogin

1.9.1 Solaris 9

Sous Solaris, le démon dtlogin assure la gestion des connexions en utilisant les protocoles XDM et XDMCP. Il présente une bannière de connexion à tout équipement qui fera la requête appropriée. Il est donc souhaitable de limiter les candidats. La copie du fichier `/usr/dt/config/Xaccess` dans `/etc/dt/config/Xaccess` suivie de son édition suffira. L'ancien fichier contient les 2 lignes non commentées suivantes :

```
*                       # grant service to all remote displays
*   CHOOSER BROADCAST   # any indirect host can get a chooser
```

Il convient de remplacer ces deux lignes par :

```
!*
!* CHOOSER BROADCAST
```

Il est également conseillé de désactiver les réponses aux requêtes XDMCP en copiant le fichier `/usr/dt/config/Xconfig` dans `/etc/dt/config/Xconfig` et en activant la dernière ligne :

```
Dtlogin.requestPort:      0
```

Il est à la suite de ces 2 opérations indispensable de relancer le démon dtlogin.

1.9.2 Solaris 10

Les versions initiales de Solaris 10 utilisent dtlogin comme sous Solaris 9 via `/etc/rc2.d/S99dtlogin`. Un patch intégré dans l'update 3 fait passer ce service sous le contrôle de SMF. L'installation sécurisée de Solaris 10 U3 sécurise le démon dtlogin en lançant `dtlogin` avec l'option `-udpPort 0`.

1.10 Réactiver certains services sous Solaris 10

Si l'on a choisi l'installation sécurisée proposée à partir de l'update 3, il est parfois nécessaire de revenir à une configuration un peu plus ouverte. X11 à travers ssh pouvant être considéré comme sûr, on l'autorisera par :

```
# svccfg -s svc:/application/x11/x11-server setprop options/tcp-listen=true
```

De la même façon, passer d'une configuration Sendmail purement locale à une configuration capable de recevoir des messages depuis l'extérieur se fera par :

```
# svccfg -s svc:/network/smtp:sendmail setprop config/local_only=false
# svcadm disable svc:/network/smtp:sendmail
# svcs svc:/network/smtp:sendmail
```

```
STATE          STIME    FMRI
disabled       12:57:38 svc:/network/smtp:sendmail
# svcadm enable svc:/network/smtp:sendmail
# svcs svc:/network/smtp:sendmail
STATE          STIME    FMRI
online         12:58:12 svc:/network/smtp:sendmail
```

2. Vérifier que la réalité correspond au cahier des charges

Même si l'on pense avoir sécurisé localement un poste de travail, il est indispensable d'en effectuer une vérification depuis un équipement tiers. Le logiciel nmap est l'outil idéal pour faire cet indispensable audit qui consiste à balayer en TCP et UDP l'équipement à auditer. Le résultat suivant sera considéré comme sûr. Les plus méfiants pourront lancer un balayage de la totalité des ports.

```
# nmap -P0 -sT rama

Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2007-04-25 15:12 CEST
Interesting ports on rama (192.168.1.3):

(The 1655 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
22/tcp  open  ssh
111/tcp open  rpcbind

Nmap run completed -- 1 IP address (1 host up) scanned in 52.781 seconds
# nmap -P0 -sU rama

starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2007-04-25 15:14 CEST
nmap rama
Interesting ports on rama (192.168.1.3):
(The 1476 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
111/udp   open  rpcbind
32771/udp open  sometimes-rpc6
Nmap run completed -- 1 IP address (1 host up) scanned in 149.094 seconds
```

3. Contrôler l'intégrité

Le système d'installation des paquets logiciels utilise une base de données contenant le type, les permissions, le propriétaire et le groupe de chaque fichier accompagné d'un total de contrôle pour les fichiers. Il est donc possible de vérifier l'intégrité des fichiers faisant partie de paquets logiciels. Il est par ailleurs assez simple de comparer la liste des fichiers suid *root* avec une liste de références. On pourra adapter le script suivant pour vérifier l'intégrité des exécutables essentiels et contrôler les fichiers suid.

```
#!/bin/ksh
# Controle d'intégrité des commandes SUID root.
export PATH=/usr/bin:/usr/sbin
DIR="/var/tmp"

function makelist
{
  df -F ufs | awk '{printf "%s\n", $1 }' | while read mp
  do
    find $mp -xdev -type f -perm -4000 -user 0 -print >> $1
  done
}

> /tmp/refsuidck.$$

### First Run

if [[ ! -f "$DIR/suid.db" ]]
then
  print "Phase 0 : Init"
  makelist "$DIR/suid.db "
  print "done."
  exit
fi

### Next run

cp "$DIR/ " /tmp/refsuidck.$$
print "Phase 1 : check integrity"
pkgchk -i /tmp/refsuidck.$$
print "done."

> /tmp/newsuidck.$$
makelist /tmp/newsuidck.$$

print "Phase 2 : check new suid"
diff /tmp/refsuidck.$$ /tmp/newsuidck.$$
print "done."

rm -f /tmp/refsuidck.$$ /tmp/newsuidck.$$

print "Phase 3 : check base package"

pkgchk SUNWcsu
pkgchk SUNWrcmds
pkgchk SUNWsshdu
pkgchk SUNWsshu
pkgchk SUNWftpu
pkgchk SUNWtftp
pkgchk SUNWipfu
print "done."
```

4. Anti-virus, anti-espion et anti-rootkit

La rareté des virus sous Solaris ne doit pas être un prétexte pour ne rien faire. Si les virus sont rares, on recense une centaine de vers et de *rootkits* capables d'affecter les systèmes Unix dans leur ensemble.

4.1 Anti-virus

Le logiciel libre Clam antivirus **[http://www.clamav.net]** communément appelé Clamav opère en général sur des serveurs

de messagerie pour éviter la propagation des virus. Il est possible de le faire travailler localement en lui demandant d'analyser des répertoires ciblés à adapter selon les outils utilisés. Il est par exemple possible d'automatiser le contrôle de certains répertoires en automatisant le lancement de clamscan.

Cela produira par exemple le résultat suivant :

```
$ cd; clamscan -i -r .mozilla .thunderbird .staroffice8 /tmp /var/tmp

produced the following output:

.mozilla/default/uuxw959q.slt/Cache/4BE588B9d01: Eicar-Test-Signature FOUND

---------- SCAN SUMMARY -----------
Known viruses: 112127
Engine version: 0.90.1
Scanned directories: 107
Scanned files: 440
Infected files: 1
Data scanned: 111.96 MB
Time: 55.629 sec (0 m 55 s)
```

La solution Avira **[http://www.free-av.com/]** est une alternative commerciale, gratuite à titre particulier, qui présente l'avantage de fonctionner en temps réel sur les flux entrants.

4.2 Anti-rootkit

Le logiciel Chkrootkit **[http://www.chkrootkit.org/]** permet de détecter si un système UNIX n'a pas été compromis par un rootkit. Ce logiciel s'installe facilement et, là encore, on automatisera son lancement.

4.3 Gestion des fichiers core avec coreadm

Solaris 9 et 10 autorisent une gestion des fichiers core via la commande `coreadm`. Cette commande configure le fichier `/etc/coreadm.conf`. Sans arguments, elle liste son contenu :

```
# mkdir /var/core; chmod 700 /var/core
# coreadm
     modèle de fichier core global :
     Contenu du fichier Core global : default
       modèle de fichier core init : core
       Contenu du fichier Core init : default
            vidages de fichier core global : disabled
       vidages de fichier core par processus : enabled
      vidages de fichier core setid global : disabled
 vidages de fichier core setid par processus : disabled
     journalisation de vidage de fichier core global : disabled
# coreadm -g /var/core/core.%f.%p
# coreadm -e global -e log -e global-setid
# coreadm
     modèle de fichier core global : /var/core/core.%f.%p
     Contenu du fichier Core global : default
       modèle de fichier core init : core
       Contenu du fichier Core init : default
            vidages de fichier core global : enabled
       vidages de fichier core par processus : enabled
      vidages de fichier core setid global : enabled
 vidages de fichier core setid par processus : disabled
     journalisation de vidage de fichier core global : enabled
```

Correctement programmé, ce système évite que les fichiers core soient écrasés par les « plantages » successifs d'un service attaqué en débordement de tampon, ce qui permet de détecter et de journaliser les tentatives d'attaques. Le service correspondant doit être activé et, sous Solaris 10, on vérifiera qu'il l'est par :

```
# svcs coreadm
STATE          STIME    FMRI
online         oct._24  svc:/system/coreadm:default
```

Sous Solaris 9, c'est `/etc/rcS.d/S42coreadm` qui lance ce service.

L'examen du fichier core peut permettre de déterminer s'il s'agit d'une anomalie ordinaire ou si on se trouve en présence d'une tentative d'attaque par débordement de tampon. Si le logiciel en cause est compilé avec l'option `-g`, on pourra utiliser le metteur au point `dbx`. Lorsqu'il s'agit d'un logiciel compilé sans cette option, l'information sera plus difficilement exploitable.

```
$ dbx /usr/lib/sendmail core.sendmail.9944.0.25.1196556767
core file header read successfully
t@1 (l@1) program terminated by signal SEGV (no mapping at the fault address)
0xfeba573c: strlen+0x000c:      movl    (%eax),%edx
```

Solaris avec les commandes `pstack`, `pflags`, `pcred`, `pldd` offre des outils permettant de mieux cerner les conditions du crash.

```
# pstack core.sendmail.16498.0.25.1196599767
core 'core.sendmail.16498.0.25.1196599767' of 16498: /usr/lib/sendmail -bd -q15m
-n
 feba573c strlen   (81a5e20, 80451fc) + c
 feebc7c3 getipnodebyaddr (8153df4, 4, 2, 80451fc) + 2a6
 08065c09 sm_gethostbyaddr (8153df4, 4, 2) + 4a
 0806be6a hostnamebyanyaddr (8153df0, 0, 8122968, d1, 8045df8, 5) + 78
 08067c19 getrequests (8135500) + ba5
 0807e574 main     (4, 8047e78, 8047e8c) + 37fd
 080605fa ???????? (4, 8047f10, 0, 8047f26, 8047f2c, 0)
```

5. Maintenir la sécurité

Au fil de sa vie, le nombre de failles connues d'un système tend à augmenter de telle sorte qu'un système non corrigé devient vite une proie facile. Il est donc essentiel d'appliquer de manière régulière les correctifs de sécurité.

5.1 Cluster de patchs

Les correctifs « recommandés » et « sécurité » sont fournis sous la forme d'un *cluster* de patchs se présentant sous la forme d'une archive zip. Cette archive nommée `9_x86_Recommended.zip` pour Solaris 9 est mise à jour plusieurs fois par mois est à appliquer au minimum 4 fois par an. Elle s'installe en mode mono-utilisateur. En admettant que l'archive zip soit chargée dans `/var/tmp`, son installation se résume à l'enchaînement des commandes suivantes :

```
# init S
# cd /var/tmp
# unzip -q 9_x86_Recommended.zip
# cd 9_x86_Recommended
# view CLUSTER_README
# ./install_cluster
```

Les recommandations contenues dans le fichier CLUSTER_README sont à prendre en compte. Si une faille de sécurité particulièrement dangereuse est à appliquer immédiatement, il faut charger le correctif et l'installer. Cela se résume par exemple à la séquence suivante :

```
# cd /var/tmp
# unzip 114145-07.zip
# view 114145-07/README.114145-07
# patchadd 114145-07
```

Selon le correctif, l'installation se fera en mode multi- ou mono-utilisateur. L'information à ce sujet est indiquée dans le fichier README du patch. Ici aussi, sa lecture est indispensable.

5.2 Solaris 10

Solaris 10 offre une évolution majeure en proposant l'utilitaire d'application de patchs signés Smpatch. À cet utilitaire de type ligne de commande est associé une interface graphique appelée Updatemanager. Ces 2 outils permettent d'enchaîner un bilan des patchs à installer, puis leur chargement, la vérification de leur signature et leur installation.

```
# smpatch analyze
...
# smpatch update -i 119283-01
```

5.3 Solaris 9

Depuis septembre 2007, Sun propose Updatemanager pour Solaris 9. Il est donc souhaitable de l'installer sur les serveurs qui possèdent les ressources pour le faire fonctionner (600 Mhz et 512 Mo est le minimum). Dans l'hypothèse contraire, on l'utilisera en mode commande ou on préférera le logiciel libre PCA [**http://www.par.univie.ac.at/solaris/pca/**]. PCA est écrit en Perl et base son analyse sur le fichier patchdiag.xref fourni par Sun. Comme Smpatch, PCA permet d'enchaîner bilan, téléchargement et installation.

```
# pca -l rms
...
# pca -i 119283-01
```

5.4 Sécurisation au niveau de la configuration système

Le fichier /etc/system permet de paramétrer la configuration du noyau Unix. Il est vivement conseillé de rajouter les lignes suivantes et de relancer le système.

```
* NFS : accept mount call from ports < 1024
set nfs:nfs_portmon=1
* Foil certain classes of bug exploits
set noexec_user_stack = 1
* Log attempted exploits
set noexec_user_stack_log = 1
```

5.5 Améliorer la journalisation syslog

La ligne suivante du fichier /etc/syslog.conf est à activer.

```
auth.notice          ifdef(`LOGHOST', /var/log/authlog, @loghost)
```

La prise en compte de la modification sera effectuée par la commande :

```
# pkill -HUP syslogd
```

La journalisation remote étant autorisée par défaut, elle sera interdite en ajoutant la ligne suivante dans le fichier /etc/default/syslogd.

```
LOG_FROM_REMOTE=NO
```

5.6 Le fichier /var/adm/loginlog

Il enregistre les échecs de login, lorsque 5 échecs successifs se produisent. Comme il n'existe pas par défaut, il est nécessaire de le créer pour déclencher la journalisation.

```
# touch /var/adm/loginlog
# chmod 600 /var/adm/loginlog
# chgrp sys /var/adm/loginlog
```

5.7 Les fichiers de configuration du répertoire /etc/default

Ce répertoire contient des fichiers de configuration permettant de modifier un certain nombre de paramètres liés à la sécurité comme ceux relatifs à la journalisation au masque des fichiers ou à la longueur minimale des mots de passe.

```
cron        inetinit    lu          nss         sys-suspend
devfsadm    init        metassist.xml passwd    syslogd
dhcpagent   kbd         mpathd      power       tar
fs          keyserv     nfs         rpc.nisd    utmpd
inetd       login       nfslogd     su          yppasswdd
```

Cette liste n'est pas complète et on peut au moins y rajouter les fichiers :

```
telnetd        sendmail        inet_type
```

5.8 Rajouter un shell à tous les comptes

Pour toutes les entrées dont le champ shell est absent, rajouter le shell `/usr/bin/nsh`. Ce fichier est le suivant :

```
#!/sbin/sh
trap '' 1 2 3 4 5 6 7 8 9 10 12 13 14 15
PATH=/usr/bin; export PATH
logger -t nsh -p auth.notice -i "Invalid login attempt from $LOGNAME"
```

Les entrées concernées sont au minimum `daemon`, `bin`, `sys`, `lp`, `gdm`, `smmps`, `uucp`, `listen`, `nobody`, `noaccess`, `nobody4`, `webservd`, `postgres`, `svctag`. On contrôlera également la cohérence du fichier `/etc/passwd` avec la commande `pwck` et celle du fichier `/etc/group` avec la commande `grpck`.

5.9 Comptes verrouillés et compte sans mot de passe

Si on affiche le fichier `/etc/shadow`, on constate qu'il existe des entrées avec mot de passe et des entrées contenant dans le champ mot de passe la valeur `*LK*` ou `NP`. La valeur `*LK*` correspond à un compte verrouillé sur lequel tout est interdit. La valeur `NP` correspond à des comptes qui ne sont pas verrouillés parce qu'ils doivent exécuter des tâches lancées par l'utilitaire `cron`. Sur de tels comptes, un fichier `.rhosts` ou une faille dans un logiciel comme telnetd ou rlogind peut permettre le login. Il est donc indispensable de verrouiller certains comptes. On évitera de verrouiller le compte `lp` et les comptes `sys` et `adm` s'ils sont utilisés. Le compte `sys` correspond à l'activation de l'utilitaire `sar` et le compte `adm` à l'activation de la comptabilité système. Sur une station, ces deux comptes peuvent donc être verrouillés sans risques.

```
# passwd -l uucp
# passwd -l nuucp
...
```

5.10 Paramètres réseau

Le fichier `/etc/default/inetinit` permet de choisir une méthode plus ou moins robuste pour générer les numéros de séquence TCP.

```
# TCP_STRONG_ISS sets the TCP initial sequence number generation parameters.
# Set TCP_STRONG_ISS to be:
#       0 = Old-fashioned sequential initial sequence number generation.
#       1 = Improved sequential generation, with random variance in increment.
#       2 = RFC 1948 sequence number generation, unique-per-connection-ID.
#
TCP_STRONG_ISS=2
```

5.11 Créer un .profile robuste

```
# .profile : mode 600

umask 066

if tty -s
then
  stty erase '^?'              # effacement caractère  ^H ou ^?
  mesg no                      # interdire les messages (talk, write, etc.)
  alias __A=$(print -n "\020")  # rappel des commandes avec keypad des
flèches
  alias __B=$(print -n "\016")
  alias __C=$(print -n "\006")
  alias __D=$(print -n "\002")
  set -o emacs                 # mode édition emacs
fi

# L'environnement de base

PATH=/usr/bin:/usr/sfw/bin:/opt/sfw/bin:/usr/local/bin:/usr/X/bin:~/bin
export PATH
MANPATH=/usr/man:/usr/sfw/man:/opt/sfw/man:/usr/local/man:/usr/X/man:
export MANPATH

export EDITOR=/usr/bin/vi

# Redéfinition de su avec transmission du shell et du magic-cookie X11
function su
{
  if (( $# > 2 )) || [[ -z $DISPLAY ]]
  then
    print "Execution de /usr/bin/su"
    /usr/bin/su $@
  else
    print "Fonction su avec positionnement de DISPLAY"
    XDISPLAY="$(uname -n)/unix:${DISPLAY#*:}"
    /usr/bin/su - ${2-root}  -c "DISPLAY=${DISPLAY} ${SHELL} -c \
      \"(/usr/X/bin/xauth -f ~$LOGNAME/.Xauthority extract - ${XDISPLAY}  | \
      /usr/X/bin/xauth merge -; ${SHELL})\""
  fi
}
```

5.12 Les logiciels tiers

S'il est finalement facile de sécuriser un poste de travail, il n'en va pas de même de la sécurisation du « trio infernal » composé du navigateur web et ses applications auxiliaires, de l'agent de messagerie électronique et de la suite bureautique. En effet, ces logiciels sont extrêmement complexes et riches en fonctionnalités, ce qui a pour conséquence d'engendrer des failles de sécurité à répétition. En outre, la fuite en avant, génératrice de nouvelles failles, est indirectement la conséquence du monopole de Microsoft sur le marché du poste de travail. En effet, il ne fait qu'aggraver la surenchère et la course aux fonctionnalités à laquelle sont poussés ceux qui ne veulent pas de ce monopole et qui s'emploient à reconquérir le terrain perdu. Microsoft ne voulant pas abandonner un pouce de terrain, la disparition des failles à répétition n'est pas pour demain.

5.12.1 Logiciels de la fondation Mozilla

Jusqu'à l'update 4 Solaris 10 est livré avec un navigateur Mozilla 1.7 régulièrement corrigé dans des délais acceptables (une mise à jour par mois en moyenne). Avec l'update 4, Sun livre en complément Firefox et Thunderbird qui ne font pas pour le moment l'objet de correctifs. Tant que cette situation n'évolue pas, il est préférable d'utiliser Mozilla ou de leur substituer les contributions Sun disponibles sur le site de la fondation Mozilla. Comme leurs mises à jour corrigent presque systématiquement des failles de sécurité, il est souhaitable de les installer régulièrement. Proposées avec la même régularité que pour les autres OS sous forme de paquets logiciels, la maintenance de Firefox et Thunderbird ne présente pas de difficultés et consiste à lancer une commande `pkgrm` suivie d'une commande `pkgadd`.

Le navigateur web est probablement le logiciel le plus vulnérable et on pourrait consacrer des dizaines de pages à la façon de l'utiliser de la manière la moins vulnérable. Il est indispensable de ne pas l'utiliser « *out of the box* » et de restreindre certaines de ses possibilités à partir des onglets suivants :

- ***Édition/Préférences/Contenu*** pour désactiver Javascript et/ou Java, bloquer les *popup* ;
- ***Édition/Préférences/Vie privée*** pour la gestion des cookies ;
- ***Édition/Préférences/Avancé*** pour la gestion des certificats ;
- ***Édition/Préférences/Sécurité*** pour la gestion des mots de passe et la détection des sites contrefaits.

On notera que tous les paramètres de configuration ne sont pas accessibles au moyen de menus, mais que l'URL spéciale **about:config** permet d'atteindre la totalité des réglages.

L'alternative aux logiciels de la fondation Mozilla consiste à utiliser la suite Opera qui offre les fonctions navigateur, messagerie électronique et carnet d'adresses. Opera est probablement le navigateur le plus rapide et celui qui respecte le mieux les standards. Il fait aussi l'objet de failles qui exigent un suivi régulier. On notera qu'il possède une fonctionnalité de téléchargement de type Torrent qu'il n'est pas prudent d'activer.

5.12.2 Plugins et extensions

Outre le langage Javascript qui existe en standard, les logiciels Firefox et Thunderbird sont extensibles par des *plugins* faisant appel à des applications externes. On compte par exemple :

- Java ;
- Flash Player ;
- Adobe Reader (Sparc seulement) ;
- Star ou OpenOffice ;
- Realplayer.

On notera que certains sont programmables à l'aide d'outils de configuration ou de fichiers de configuration. La version 9 de flash utilise le fichier `/etc/adobe/mms.cfg` pour modifier les paramètres de sécurité. Adobe indique comment configurer ce fichier dans le document de référence `flash_player_9_security.pdf`.

Par ailleurs, il existe désormais une nouvelle variété d'ajouts appelés extensions s'installant par le chargement d'un fichier d'extension `.xpi`. On recense parmi les milliers de possibilités des logiciels réellement très utiles parmi lesquels on peut citer :

- NoScript **[https://addons.mozilla.org/fr/firefox/addon/722]** pour un filtrage du Javascript selon le site visité ;
- QuickJava **[https://addons.mozilla.org/fr/firefox/addon/1237]** pour une activation rapide tout ou rien et distincte de Java et/ou de Javascript ;
- Enigmail **[http://enigmail.mozdev.org/]** pour le chiffrement et la signature PGP des messages.

La mise à jour de ces logiciels auxiliaires est aussi importante que celle du navigateur. On notera qu'elle peut être automatisée.

5.13 Suites bureautiques OpenOffice et StarOffice

StarOffice 7 est livré en standard sous Solaris 10 avec un niveau de correction dépendant de l'update utilisé. Il sera corrigé à l'aide du système de patch standard ou remplacé par StarOffice 8 qui devra subir le même traitement. Cette possibilité est l'un des avantages de StarOffice sur OpenOffice. En effet, pour ce dernier, une mise à jour se traduit par une suppression suivie d'une installation y compris celle des dictionnaires et extensions qui sont venues en ajout.

Open et StarOffice proposent un système d'extensions assez comparable à celui de Firefox et il est tentant de compléter OpenOffice. Comme pour Firefox, on contrôlera régulièrement la disponibilité de mises à jour via l'onglet ***Outils/Gestionnaire des extensions***, si on utilise cette possibilité.

Les réglages accessibles depuis l'onglet ***Outils/Options*** sont à contrôler en particulier pour :

- ***OpenOffice.org/Sécurité*** : s'assurer que les paramètres accessibles depuis ***Sécurité des macros*** ne sont pas trop permissifs ;
- ***OpenOffice.org/Java*** : contrôler que la version activée n'est pas vulnérable ou désactiver Java ;
- ***OpenOffice.org/Mise à jour en ligne*** à paramétrer selon la réactivité souhaitée ; on peut aussi utiliser l'onglet ***Aide/Rechercher des mises à jour*** pour tenter le chargement ;
- ***Internet/Proxy*** permet de désactiver les accès Internet ;
- ***Internet/Plugin Mozilla*** selon la nécessité d'activer StarOffice comme plugin de Firefox.

L'application sans délai des mises à jour corrigeant les failles de sécurité est indispensable lorsqu'on utilise OpenOffice comme plugin du navigateur pour les fichiers au format OpenDocument tout comme ceux utilisant le format MicroSoft Office.

Supervision et sécurité par analyse des flux

Les technologies généralement déployées pour la supervision réseau reposent sur une interprétation des données contenues dans les paquets. Elles sont efficaces, mais ont toutefois le désavantage d'être consommatrices en ressources, de nécessiter des signatures élaborées et surtout d'être inopérantes sur des protocoles chiffrés. Ces inconvénients rendent la recherche de signature dans les paquets IP pertinente sur un Intranet, mais de moins en moins réaliste sur des points d'accès Internet conséquents. Leur efficacité est également limitée lorsqu'il s'agit de réaliser une analyse a posteriori de l'activité réseau, en vue d'une détection ou de l'analyse détaillée d'une compromission. A contrario, l'analyse des flux collectés pour la métrologie n'est pas tributaire d'une bande passante ou de protocoles particuliers. Par contre, l'interprétation de ces flux à des fins de supervision sécurité peut s'avérer assez fastidieuse sans un post-traitement adapté qui vise à passer du flux à l'alerte. Cet article présente les quelques pistes suivies par le CEA dans ce domaine, ainsi que son retour d'expérience sur les résultats obtenus.

mots clés : *représentation / trafic / attaques / détection d'intrusion / métrologie*

Les limites atteintes

Notre engouement pour la recherche de signature et pour l'inspection de contenu dans la supervision réseau a quelque peu diminué devant des techniques d'évasion qui nécessitent des protections de plus en plus complexes et délicates à maintenir et surtout devant la généralisation du chiffrement sur les réseaux WAN. Notre retour d'expérience sur certaines analyses *forensiques* a d'ailleurs démontré l'absence complète d'alertes de sécurité dans les outils de supervision à base de signatures et a confirmé la nécessité de conserver les flux réseau de manière exhaustive sans se limiter aux seuls flux comportant une signature connue.

A contrario, la collecte d'information sur les flux réseau WAN nécessite beaucoup moins de ressources (y compris en termes de capacités de stockage) et a l'avantage d'être exhaustive dans sa collecte d'informations. Le niveau de détail d'un flux peut être assez varié en fonction des capacités des équipements de collecte, mais on s'intéresse généralement aux informations telles que définies dans le protocole Netflow de Cisco [1] comme la source, la destination, les informations horaires (début de session, fin de session, durée), la volumétrie (nombre de paquets) et les informations protocolaires (protocole, port source et destination, flags des paquets). De manière schématique, on sait « qui communique avec qui, quand et comment », mais on ne garde aucune trace du contenu de la conversation.

Si la métrologie réseau peut se contenter du flux, la supervision sécurité est cependant moins immédiate, car un flux ne constitue pas en soi une alerte de sécurité (mis à part le cas trivial où celui-ci est en violation de la politique de filtrage). Ceci suppose que pour identifier dans l'ensemble des flux réseau les scénarios inquiétants ou anomalies susceptibles de générer une alerte ou un incident de sécurité, il va falloir mettre en œuvre des outils de post-traitement, comme la suite d'outil Netsa [2], qui vont rendre illusoire une analyse en temps réel. À la différence du paquet qui déclenche une signature et une alerte quasi immédiate, il faut également se rappeler que le flux est une information qui n'est globalement émise qu'à la fin de l'activité réseau. Il n'est par conséquent, pas adapté à une réaction en temps réel. Est-ce vraiment gênant ?

Pas vraiment dans notre cas. D'une part, on ne cherche pas à concurrencer les IPS en production, d'autre part, l'engouement pour la supervision temps réel ne correspond pas à une réalité d'exploitation, notre objectif étant davantage une analyse en temps différé réalisée à une fréquence régulière (le rapport de la veille, du mois écoulé, etc.).

L'analyse de flux est une approche complémentaire qui ne remet pas en cause les outils existants, mais qui cherche à aiguiller et à faciliter l'analyse humaine a posteriori, en développant des outils qui ne génèrent qu'une faible charge d'exploitation supplémentaire et qui vise à fournir à la manière du *data mining*, une autre façon de visualiser les données existantes.

Mise en place d'une base de connaissances

Caractériser un flux réseau pour en analyser le danger potentiel, suppose d'enrichir les informations liées à une adresse source ou destination. L'objectif de notre base de connaissances est de constituer un référentiel destiné à fournir ces informations qui vont permettre d'apporter un éclairage pertinent sur une analyse de flux. Il est entendu que cette collecte d'informations ne doit pas solliciter de charge supplémentaire d'exploitation et qu'elle se doit d'être automatique et réactualisée.

La clé sur laquelle s'appuie la base de connaissances est l'adresse IP ou plus exactement un intervalle d'adresses IP (IP Range) caractérisé par une adresse de début et une adresse de fin. Autour de cet ensemble d'adresses, des processus automatisés réguliers alimentent des informations suivant des catégories variées comme :

- les ports TCP ou UDP ouverts tels que détectés par des scanneurs de ports (utilisation de Nmap) ;
- la prédiction du système d'exploitation utilisé (fournie également par Nmap) ;
- les informations relatives à la localisation d'une adresse IP (Pays, Ville, Latitude et Longitude) en utilisant les sources d'informations comme Maxmind [3] ;

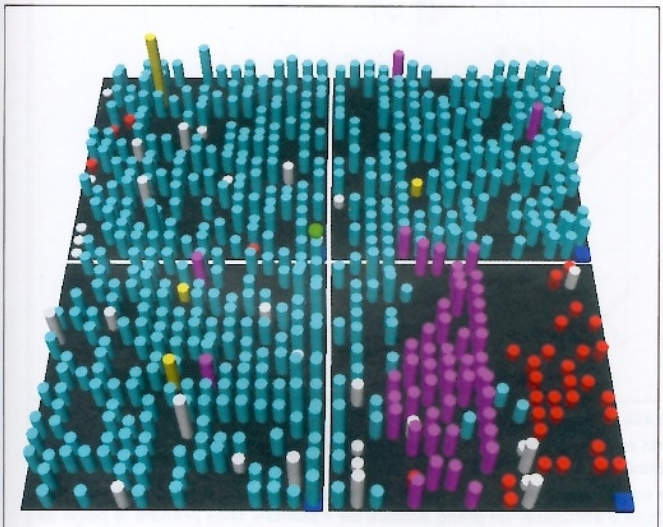
Jean-Philip Guichard – jean-philip.guichard@cea.fr,
Cédric Vieau – cedric.vieau@cea.fr,
Jean-Marc Zuccolini – jean-marc.zuccolini@cea.fr

- les informations relatives au propriétaire des adresses concernées (organisme) en utilisant les bases de registre comme RIPE, APNIC, mais aussi les listes de blocs comme Peerguardian [4].
- Les vulnérabilités associées à une adresse IP telles que signalées par Nessus.

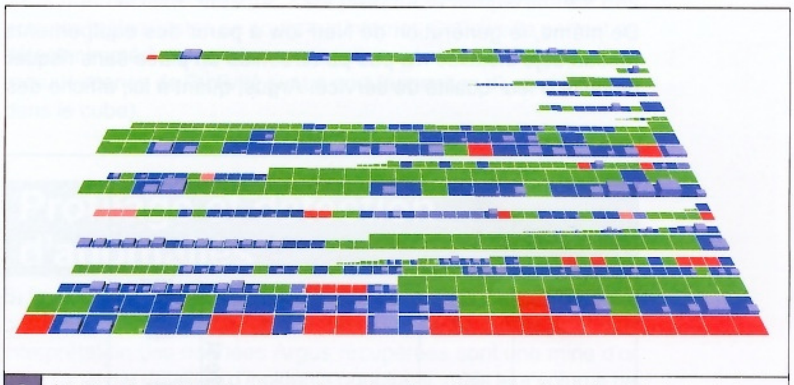
La base de connaissances est également complétée par les différents profils de sécurité réseau internes à l'organisation pour savoir si une ressource appartient à l'Intranet, à la DMZ ou encore à un *hotspot*. Cette base constitue une table unique Postgres qui contient actuellement 2,5 millions d'enregistrements. Les opérations de mise à jour et de consultation restent performantes grâce à l'utilisation de l'extension IP4R [5] qui permet une indexation efficace sur les adresses et réseaux de type IPv4. La clé de cette table étant un intervalle réseau constitué d'une adresse IP de début et d'une adresse IP de fin, elle permet de limiter le nombre d'enregistrements, puisqu'il est possible d'utiliser un mécanisme d'héritage où les petits réseaux héritent des attributs des réseaux les incluant.

En l'état, l'exploitation de la base de connaissances s'apparente à celle d'une base de type inventaire et demeure relativement aisée lorsqu'on sait précisément ce que l'on recherche : statistiques sur le parc déployé, recherche du propriétaire d'une machine, liste des systèmes vulnérables à une nouvelle attaque sur un port déterminé, etc. En revanche, compte tenu du volume, une exploration globale de la base est difficilement réalisable par des outils de *reporting* classique SQL.

Pour exploiter au mieux ces informations, un mode de représentation graphique en 3D a été développé. L'outil est conçu pour être utilisé en ligne à partir d'un navigateur web et il repose pour cela sur du code PHP qui génère des vues VRML (*Virtual Reality Markup Language* est un langage de description normalisé d'univers virtuels en 3D).



1 *Résultat d'un scan Nmap sur un réseau bureautique composé en majorité de postes Windows (bleus), avec des imprimantes (violettes) et quelques équipements réseau (rouges). Les machines dont l'OS n'a pas été identifié (blanches), ainsi que quatre postes Linux (jaunes) attirent notre attention, surtout celles dont la hauteur du cylindre montre qu'elles possèdent de nombreux ports ouverts.*



2 *Lorsque le nombre de réseaux à représenter est important (ici l'équivalent de plusieurs classes B), le mode de représentation change : chaque réseau est surmonté d'un cube dont la hauteur est le maximum des hauteurs des cylindres qu'il contient, et la superficie de la base indique le taux d'occupation du réseau. Les couleurs sont choisies en fonction du profil de sécurité du réseau (extranet, DMZ, etc.). Lorsque l'on zoome sur cette carte, l'affichage détaillé en cylindres apparaît.*

Les machines sont regroupées par sous-réseaux et représentées sous la forme de cylindres dont la couleur dépend du système d'exploitation, et la hauteur varie en fonction d'indicateurs sur les ports ouverts : nombre total de ports, nombre de ports « Web », nombre de ports atypiques ou vulnérables, etc.

Globalement, la vision synthétique qu'offre cet outil permet de naviguer dans l'équivalent d'un réseau de classe B de manière confortable et de faire varier une dizaine d'indicateurs faisant évoluer la hauteur des colonnes.

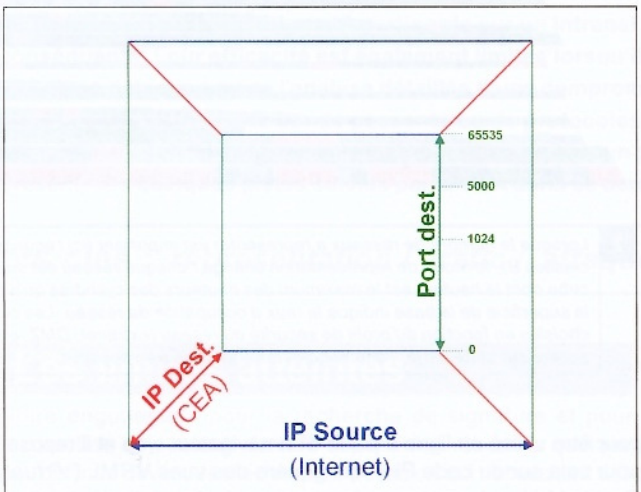
Analyse des flux

L'analyse des flux réseau a plusieurs applications pratiques : détection de compromissions, de violation de politique, de prolifération d'un vers, cartographie, etc. (Cf. MISC 17 « Les flux réseau » par Nicolas FISCHBACH [6]). Elle offre dans la supervision de nombreux avantages : d'une part, elle n'est pas discriminatoire (conserver un flux réseau, c'est s'assurer d'être exhaustif et factuel), d'autre part, elle n'est pas tributaire d'une règle particulière.

Notre analyse de flux est mise en œuvre autour de sondes passives placées sur les accès Internet, en amont de tous les dispositifs de protection et de filtrage. Ce choix s'explique pour deux raisons principales : d'abord, nous ne visions pas une analyse de contenu des paquets (pour éviter notamment toute redondance avec les équipements actifs déjà mis en place derrières les pare-feu). Si tel avait été le cas, il aurait été suicidaire en termes de charge requise pour une analyse de contenu de placer la sonde « toute nue » sur Internet. Ensuite, sur le principe d'exhaustivité, notamment lié à l'exercice d'analyse forensique, il est intéressant de récupérer tout le trafic en provenance d'Internet, même s'il est ensuite bloqué par les équipements actifs.

Pour recueillir les flux réseau, la technologie choisie a été le logiciel Argus [7]. D'autres outils d'écoute passive sont disponibles, mais le choix s'est fait principalement sur la capacité à tenir la charge WAN. Ainsi, des outils comme OSSIM [8] ou Ntop [9] ont dépassé nos capacités de traitement (charge CPU trop élevée).

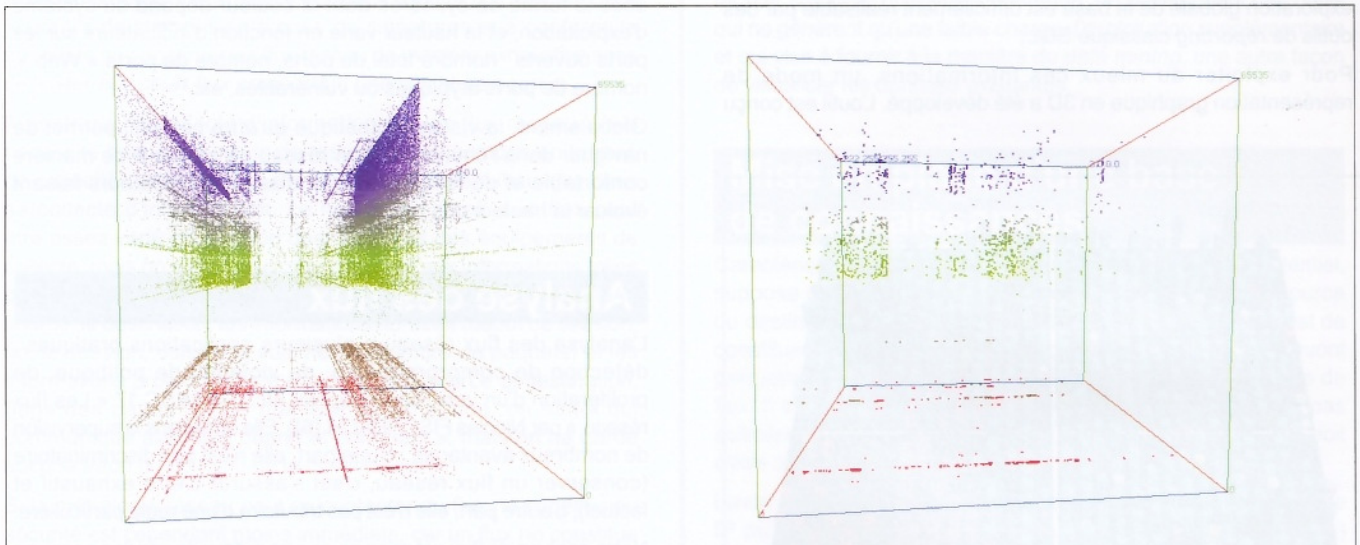
De même, la génération de NetFlow à partir des équipements réseau déjà installés n'a pas pu être mise en place sans risquer de nuire à leur qualité de service. Argus, quant à lui, affiche des performances surprenantes sur une configuration matérielle à peine gonflée (biprocesseur Xeon 3 GHz, 4 Go Ram) au vu des débits réseau gérés (liens Gb). La perte de paquets est vraiment minime et aucun souci de performance machine n'est à déplorer.



3 *Légendes des axes*

Argus dispose aussi d'un avantage important par rapport aux autres collecteurs de flux classiques. Il s'intéresse davantage aux sessions qu'aux flux à proprement parler. La technologie Netflow suppose que pour une session classique client-serveur, deux enregistrements soient produits pour chaque sens de communication (Par exemple : un pour le flux du navigateur vers le site et un autre pour la réponse du site au navigateur). La reconstitution de la session à partir des deux flux générés est de la responsabilité du collecteur qui se base pour cela soit sur une interprétation des flags dans le cas d'un protocole connecté comme TCP, soit sur une interprétation de la communication basée sur des fenêtres de temps pour associer le flux descendant et montant. Des évolutions du format IPFIX et la spécification d'un mode *biflow* [10] permettront à la sonde de prendre en charge cette analyse et de pouvoir la communiquer aux collecteurs, mais elles sont encore à l'état de *draft*. Cependant, il s'agit du fonctionnement natif d'argus de reconstituer le sens de la communication et c'est une information d'une part primordiale pour une analyse sécurité, d'autre part, très délicate à reconstituer au niveau des collecteurs.

Argus est un outil de capture de flux réseau (par opposition à la capture de paquets) qui reconstitue les sessions, c'est-à-dire que pour une connexion TCP entre deux machines, un seul enregistrement sera créé, et il contiendra le nombre de paquets échangés, les dates de début et fin de connexion, le volume de données transmises, etc.



4 *L'image de gauche montre une vue du cube après deux heures d'activité accumulée : ~150000 flux, établis ou non.*
On remarque plusieurs « murs » de connexions (rectangles verticaux violets/verts en haut du cube) : chacun d'eux représente un client d'Internet qui tente de se connecter sur de nombreuses machines du CEA, et à destination de l'ensemble de leurs ports.
Ce type de scan dure généralement plusieurs jours, et l'ordre des tentatives de connexion semble aléatoire. Il s'agit typiquement d'une méthode de scan très difficile à repérer avec les outils traditionnels ; par contre, le cube a la capacité d'accumuler un grand nombre de « points » sans prendre la peine de les comprendre, l'œil humain se chargeant de détecter les motifs qui apparaissent. On note aussi deux « plateaux » sur la moitié inférieure du cube : un rouge (port 139) et un marron (port 445). Ils représentent l'activité de propagation virale de l'univers Windows. On remarque que leur provenance n'est pas uniforme sur l'espace d'adressage d'Internet : certaines plages réseau sont plutôt occupées par des entreprises et autres organismes ayant mis en place des protections réseau, et les plages les plus denses en virus sont principalement attribuées au grand public…
À droite, seuls les flux établis sont représentés. Les lignes rouges horizontales du bas (parallèles à l'axe bleu) représentent l'ensemble des clients d'Internet accédant aux serveurs publics du CEA. On peut également observer des connexions établies depuis Internet sur des ports « hauts » (points verts et violets), correspondant aux flux « data » du protocole FTP en mode « passif ».

Ne conserver que les flux augmente la capacité de stockage des journaux de manière significative. Sur des liens qui voient passer jusqu'à 30.000 paquets par seconde, pour des débits allant jusqu'à 800 Mbs, les journaux Argus générés ne représentent que 4 Go compressés par jour (~12 Go bruts). Ce volume nous permet de conserver pendant plusieurs mois l'intégralité des flux échangés entre notre réseau et Internet, ainsi que l'ensemble des tentatives d'attaque subies.

Visualisation temps réel : le Cube

Même si l'orientation du projet ne visait pas à s'investir dans la détection d'attaques en temps réel, il était tentant de pouvoir visualiser l'activité courante vis-à-vis d'Internet. Nous inspirant du *Spinning Cube of Potential Doom* de Stephen Lau [11], nous utilisons une représentation graphique en forme de cube des logs générés par la sonde Argus. Le cube représente l'activité entrante sur le CEA depuis Internet. Les trois dimensions du cube sont utilisées pour représenter les adresses IP sources (Internet), les adresses IP destination (notre réseau), et les ports destination (ou source). Voir les figures 3 et 4, page précédente.

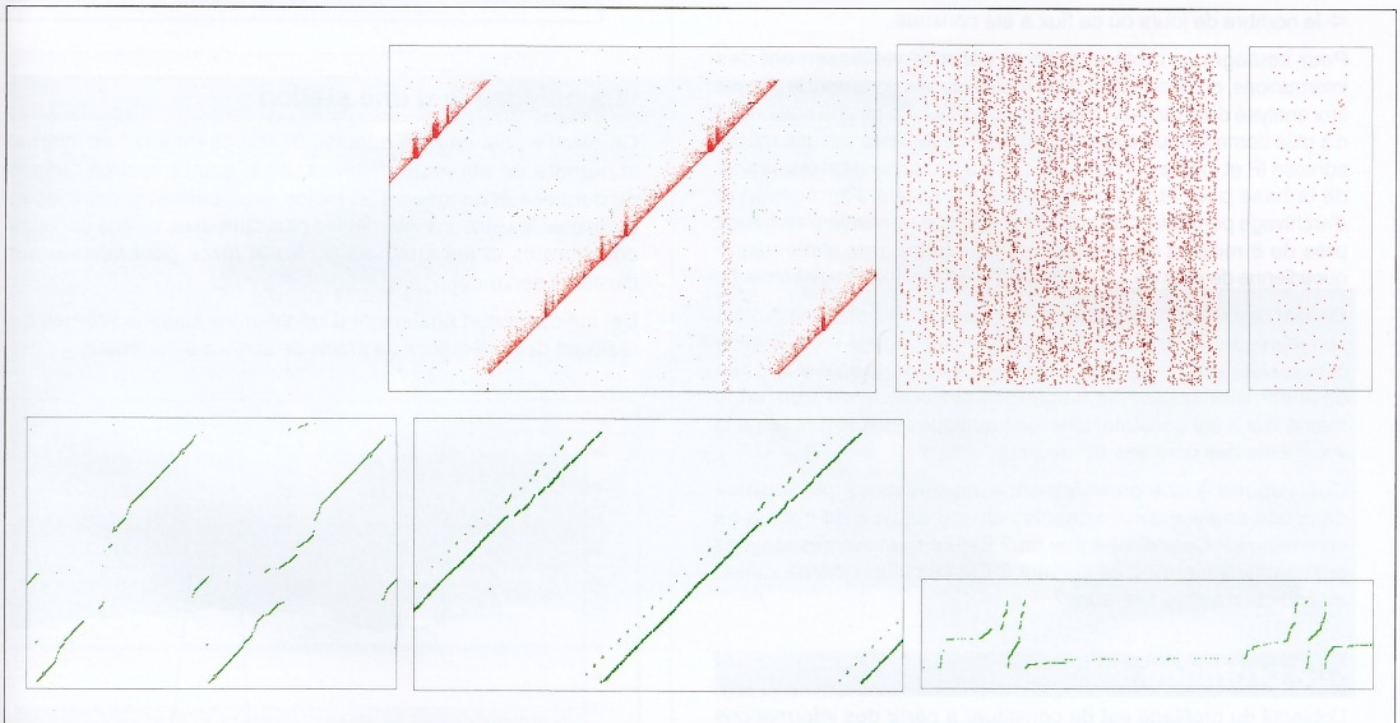
Nous voyons dans cette représentation originale des logs un véritable outil de sensibilisation à la menace internet. En effet, nous ne pensions pas Internet à ce point « virulent ». Moins de 3% : c'est le ratio entre le nombre de flux licites et le nombre de tentatives de connexions reçues. Ainsi, 97% des communications vers nos réseaux sont du bruit Internet. Dans ce bruit, il est très facile de détecter une activité de type Scan de ports (horizontal/vertical). En configurant les temps d'affichage des points à l'intérieur du Cube, on peut déceler des scans étalé « furtivement» sur plusieurs heures, voire plusieurs jours. Il est également facile de détecter les activités *peer to peer* qui se caractérisent facilement par cette représentation graphique.

Profitant de l'espace d'adressage conséquent alloué au CEA (plusieurs réseaux de classe B), le cube nous permet également, dans une certaine mesure, « de prendre la température » des activités malveillantes en cours sur Internet. Cette veille permet de détecter immédiatement les propagations virales massives par une recrudescence de l'activité sur un port (apparition d'un « plateau » dans le cube).

Profilage et détection d'anomalies

Si le stockage des fichiers de capture quotidiens d'argus ne pose pas de problème d'archivage, il n'en est pas de même pour leur interprétation (les données Argus récupérées sont une mine d'or pour l'analyse détaillée d'incidents ponctuels, mais leur volume ne permet pas de consulter ou de traiter facilement ces informations sur une longue période de temps).

Pour faciliter l'analyse des données Argus sur une grande période, il a été choisi de stocker en base de données l'ensemble des flux établis sur une période glissante d'au moins un mois. Contrairement à la vision offerte par le cube qui confronte les tentatives aux flux aboutis, on ne décide de conserver en base que les flux établis. Ce dernier volet du projet supervision appelé PITRE (Profilage et Interprétation du Trafic Réseau Étendu) consiste donc à alimenter une base de données évènementielle à partir des flux argus, et de confronter ces évènements à la base de connaissances pour identifier des anomalies de fonctionnement.



5 *Les scans horizontaux (un même port testé sur n machines) sont représentés dans le cube par une ligne horizontale parallèle à l'axe rouge. L'outil permet de modifier l'axe vertical pour qu'il utilise le port source à la place du port destination. Ce mode de représentation montre une variété étonnante d'ordonnancement des scans et de stratégies d'allocation des ports sources. Une étude plus approfondie de ces signatures permettrait peut-être d'identifier les outils ou les méthodes de scan utilisées.*

La méthode d'agrégation et d'archivage

La principale difficulté dans l'intégration des flux réseau en base de données fut le choix de la méthode d'agrégation. En effet, avec une moyenne de 1,5 millions de nouveaux flux quotidiens, nous n'avions pas la capacité de gérer une accumulation simple et chronologique. L'agrégation permet d'atteindre un nombre et un volume d'enregistrements raisonnables tout en garantissant une relative stabilité de la base évènementielle dans le temps. La perte inhérente d'information est acceptable, car il est toujours possible de se référer aux données brutes d'Argus autant que de besoin (à l'usage on se rend compte que les données en base sont bien souvent suffisantes, et que les données Argus ne servent que très rarement).

La méthode d'agrégation choisie pour les flux en base de données consiste à ne considérer comme clé unique que les informations suivantes : Protocole ; Adresse IP Source ; Adresse IP Destination ; Port Destination.

Nous ne tenons pas compte du port source jugé peu pertinent dans l'analyse.

Autour de cette clé unique qui décrit généralement l'accès depuis un client à un service donné, on agrège les informations suivantes :

- la première fois que ce dialogue a eu lieu (*timestamp* du début premier flux réseau) ;
- la dernière fois que ce dialogue a eu lieu (*timestamp* du début et de la fin du dernier flux réseau) ;
- les compteurs liés au volume échangé (nombre de paquets en entrée/sortie, nombre d'octets en entrée/sortie pour le dernier flux et en cumulé pour l'ensemble des flux) ;
- le nombre de jours où ce flux a été constaté.

Pour soulager la table activité et gérer le vieillissement des informations, on décide d'archiver toutes les sessions dont le dernier flux analysé date de plus d'un mois. Ceci signifie qu'une station qui n'a plus communiqué avec un serveur (c'est-à-dire sur une même adresse IP et même port) depuis un mois voit sa session disparaître de la base pour être archivée. Ces méthodes d'agrégation et d'archivage permettent de conserver en ligne de manière stabilisée près de 8 millions d'enregistrements, malgré une alimentation quotidienne de près de 1,2 à 1,8 millions nouveaux enregistrements.

On dispose donc d'une table de l'ensemble des stations qui ont communiqué suivant un protocole et un port donné à destination d'un serveur avec la date de la première session, la date et la durée de la dernière session, le nombre de sessions et de jours où le même flux a été constaté, ainsi que quelques indicateurs liés à la volumétrie des données échangées.

Ceci répond à une première série de questions récurrentes dans une analyse d'évènements : Quand ce type de trafic a-t-il commencé ? Quand a-t-il pris fin ? Est-ce que d'autres stations ont communiqué avec ce serveur ? Est-ce qu'une même station a contacté d'autres serveurs ?

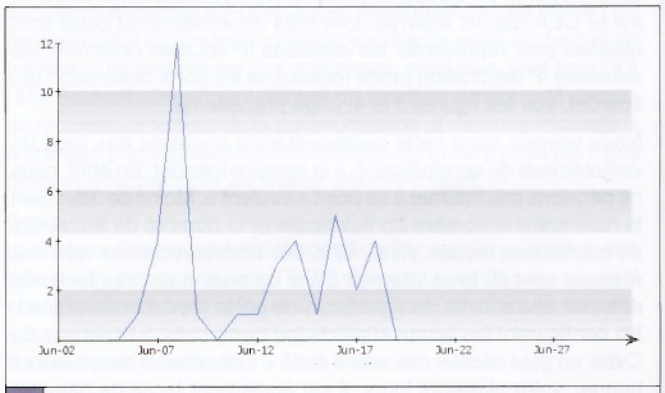
Le profilage

L'objectif du profilage est de constituer à partir des informations en base, essentiellement la base de connaissances et la base des flux, la carte d'identité d'une adresse IP basée sur ce que l'on sait d'elle a priori et sur son comportement réel. À partir de ces caractéristiques, on peut espérer pouvoir comparer l'activité future pour identifier des changements suspects. Le profilage repose sur un certain nombre d'indicateurs « maison » que l'on calcule au jour le jour et qui viennent compléter les indicateurs classiques de métrologie (bande passante utilisée, taille de paquets).

Dans la supervision réalisée par l'outil, ce qui qualifie la pertinence d'un indicateur en termes de sécurité, c'est moins sa valeur instantanée que son évolution dans le temps.

La popularité d'un serveur/service

On calcule pour chaque destination (Internet ou interne) le nombre de clients distincts, ainsi que le volume cumulé de données échangées. L'évolution de cet indice dans le temps nous permet d'une part de vérifier la bonne tenue de nos sites web, mais aussi de détecter des services non conformes à la politique de sécurité réseau.

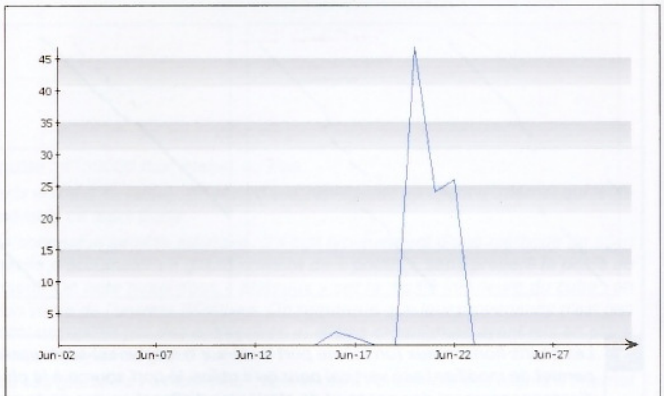


6 *Évolution de la popularité d'un serveur IRC prenant le contrôle de postes infectés.*

La polygamie d'une station

On calcule pour chaque adresse IP source (Internet ou interne) le nombre de serveurs distincts, ainsi que le volume cumulé de données échangées. Cet indice nous permet d'identifier en particulier les stations internet les plus curieuses de nos serveurs collaboratifs et les attaques en *brute force* généralisées sur plusieurs serveurs.

Cet indice permet également d'identifier les stations internes qui réalisent des opérations de scans de service à l'extérieur.



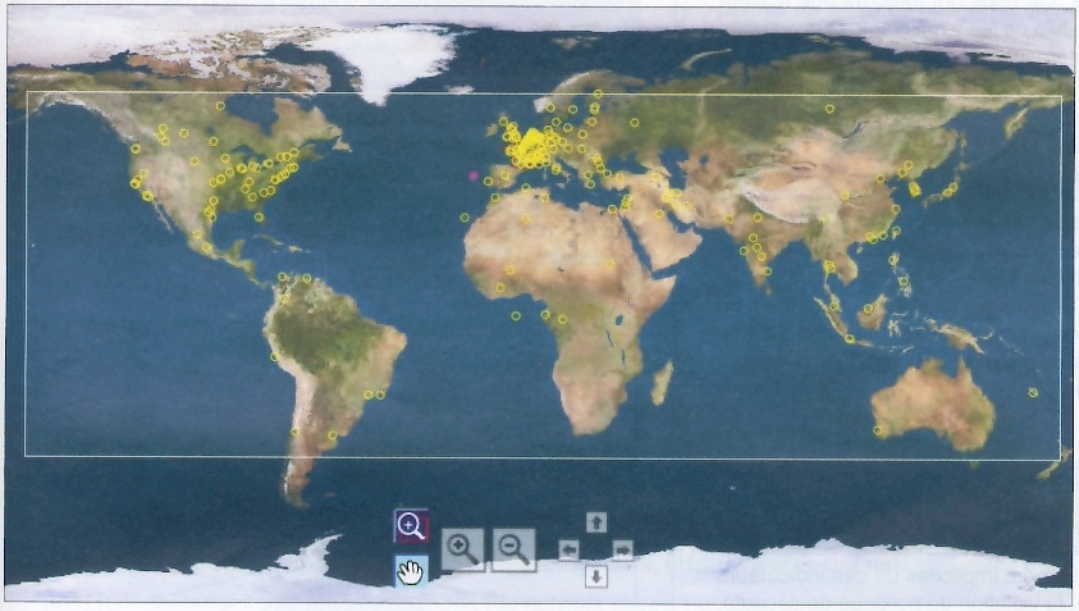
7 *Évolution de la polygamie d'un des postes enrôlés par le serveur IRC*

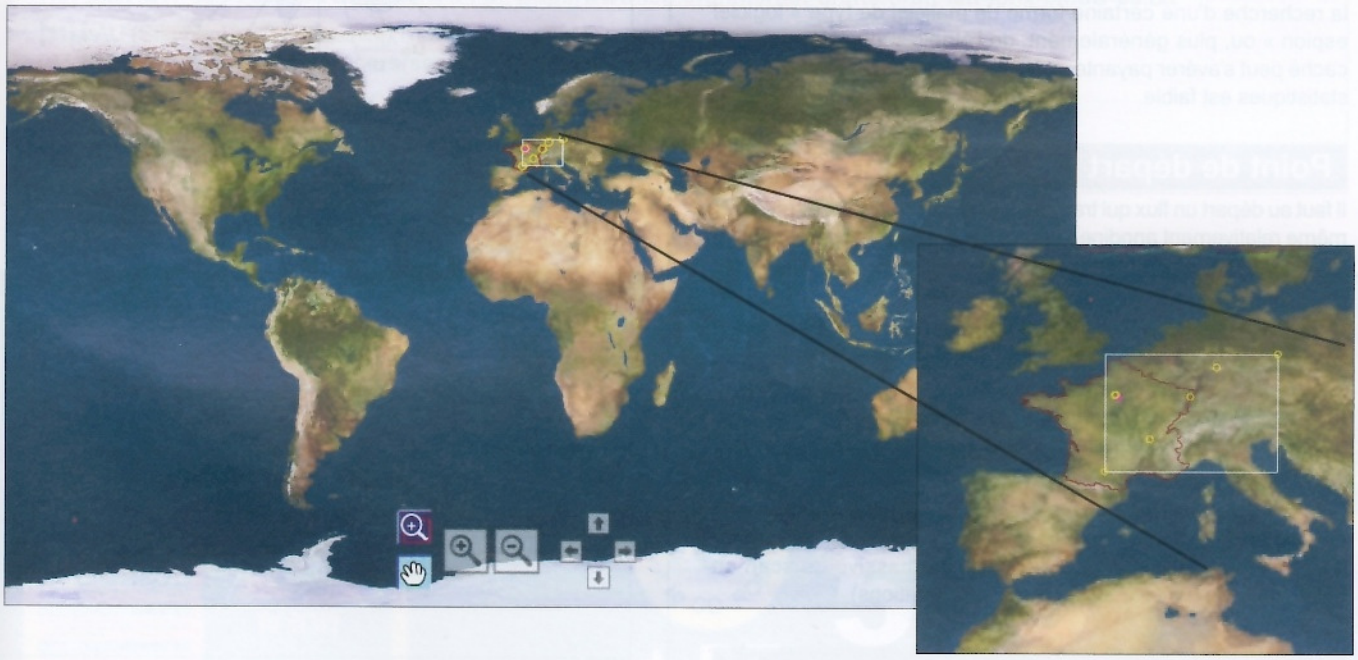
La couverture géographique d'un serveur

On utilise les fonctions géospatiales de Postgres (avec l'extension Postgis [12]) pour calculer des surfaces ou des distances à partir des différents interlocuteurs d'une adresse IP dont la latitude/longitude est connue. On calcule ainsi la surface géométrique de l'ensemble de ces interlocuteurs que l'on représente sur une carte du monde [13].

On peut ainsi constater qu'un serveur peut être très populaire (beaucoup d'IP distinctes), mais que tous ses interlocuteurs sont relativement proches géographiquement à l'échelle du globe. La variation de la couverture géographique de l'activité traduit un risque d'intrusion dans une communauté établie.

L'intérêt des fonctions géospatiales est de pouvoir passer facilement d'un mode de représentation visuel à un mode de traitement





8 *Exemples de géolocalisation des clients de serveurs du CEA : dans le premier exemple, le serveur est contacté par une grande diversité de clients. Il s'agit d'un site Web pour lequel aucun profil de nationalité ne peut être établi pour ses clients. Par contre, dans le second exemple, on constate que les clients proviennent exclusivement de la zone européenne. Il s'agit ici d'un serveur utilisé dans le cadre d'une collaboration européenne : en suivant l'évolution de l'origine des clients, il sera possible de détecter une éventuelle compromission.*

mathématique plus adapté à des moteurs d'alertes (évolution d'une distance, calcul de barycentre ou de surface).

Les amis de longue date et les nouveaux amis

Le mode d'agrégation des flux choisi a pour effet de bord de mettre en évidence les communications de très longue durée. Lorsqu'il s'agit de stations qui établissent de manière quasi quotidienne un flux à destination d'un serveur particulier, elles ne sont jamais purgées de la table des flux, car elles ont toujours une activité minimale dans le mois qui justifie de les y maintenir. Ces activités de longue date sont aussi intéressantes, car il peut arriver qu'en activité sortante sur Internet, elles traduisent un comportement de type *spyware* ou maliciel.

Le profilage temporel

Une autre signature est apparue comme utile et significative dans l'analyse de flux : il s'agit de la répartition de l'activité d'une IP dans une journée de 24h ou sur une semaine. Ce profil permet d'établir de manière grossière, mais assez significative, la présence d'un humain derrière l'activité réseau, ce dernier se trahissant par des périodes d'inactivité liées à ses besoins physiologiques primaires comme celui de se reposer le week-end.

Exercice pratique : trouver le maliciel

Outre les applications relativement implicites de ces indicateurs comme l'analyse forensique, la violation de politique de sécurité, l'apparition d'un nouveau service pirate ou une propagation virale, la recherche d'une certaine forme de maliciel de type « logiciel espion » ou, plus généralement, de l'établissement d'un tunnel caché peut s'avérer payante, même si la variation des indicateurs statistiques est faible.

Point de départ : un flux surprenant

Il faut au départ un flux qui traduise une évolution comportementale, même relativement anodine, dans la communication d'une station. Cela peut se traduire comme l'apparition d'un nouveau protocole ou d'une nouvelle destination, d'une activité nocturne atypique, d'un balayage systématique sur plusieurs serveurs ou encore d'une alerte préalablement déclenchée par un IDS.

Dans le cas le plus défavorable, la variation est vraiment anodine comme un nouveau serveur web accédé en HTTPS depuis un poste interne en journée. Il faut alors quelques éléments aggravants pour renforcer la présomption d'incident.

On évitera le cas trivial où le nouveau serveur est déjà identifié comme offensif dans la base de connaissances (connu lors d'incidents précédents, pour son activité massive de scan ou disposant d'un nom DNS traduisant ses intentions).

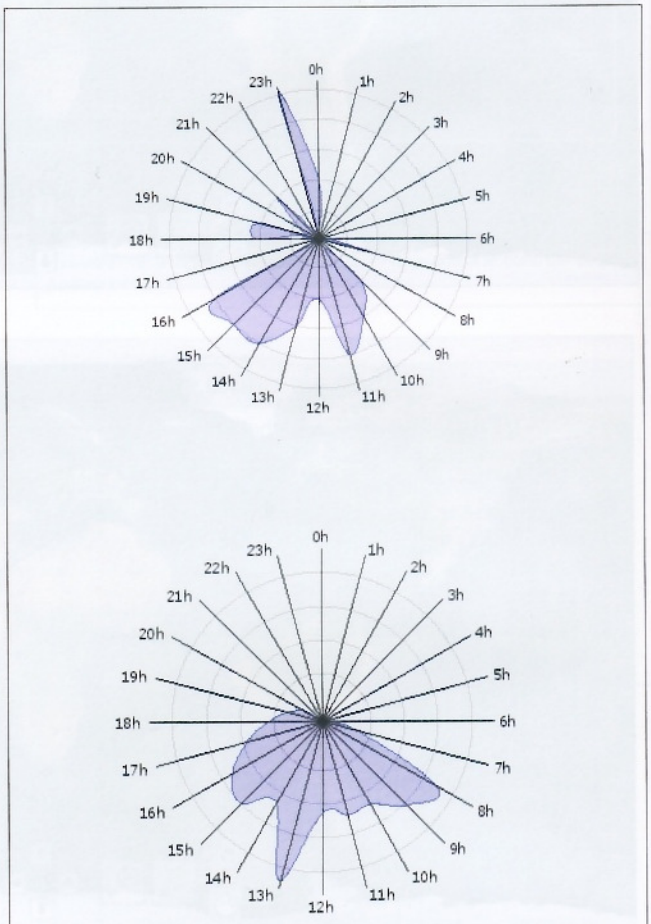
Les éléments aggravants : ce flux perturbe les caractéristiques connues

Les perturbations peuvent être de plusieurs natures. À la différence de la signature d'un IDS, l'analyse de flux ne permet de constituer au mieux qu'un faisceau de présomptions. Dans l'exemple choisi, c'est davantage l'accumulation d'anomalies qu'une anomalie en particulier qui risque de déclencher une alerte. Qu'est-ce qui fait de ce nouveau serveur HTTPS, un interlocuteur potentiellement inquiétant ? Plutôt que de chercher à évaluer la dangerosité des nouveaux sites, on cherche à retrouver le site dont l'apparition coïncide avec un changement de comportement des clients (géolocalisation des serveurs accédés, activité nocturne soudaine, hausse de la polygamie…).

Autre élément aggravant significatif, heureusement plus facile à détecter, c'est lorsque ce changement de comportement est partagé par un nombre croissant de clients internes.

Faux positif : la désillusion ?

Quel est ce nouveau service inconnu jusqu'alors, qui intéresse plusieurs dizaines de clients pendant 3 jours et qui disparaît, pour être remplacé par un autre service tout aussi populaire ? Un serveur d'enrôlement de zombies ? Non, un serveur de bandeaux publicitaires.



9 *En haut, signature horaire d'un serveur suspect utilisé dans la mise en œuvre d'un canal caché.*
En bas, signature horaire d'un serveur suspect, mais dont l'analyse a posteriori a démontré qu'il s'agissait d'un serveur de bandeaux publicitaires.

Les serveurs de bandeaux publicitaires présentent des caractéristiques assez troublantes. Ils utilisent des protocoles très répandus, ils sont inconnus la veille, populaires le lendemain et très volatiles en termes de positionnement sur Internet. La seule différence que nous avons pu constater avec des outils de type maliciel réside dans le fait que les flux liés aux bandeaux publicitaires sont inhérents à l'activité interactive de l'utilisateur, contrairement aux maliciels qui présentent une activité autonome qui dérive sur le profil horaire du poste infecté.

La cerise sur le gâteau

Comme on peut le constater, l'analyse de flux suppose la mise en œuvre de règles assez complexes pour passer de la collecte à la génération d'alertes pertinentes. Cette analyse se rapproche quelque peu des modèles d'analyse comportementale ou statistique, bien qu'elle ait la chance d'être plus facilement modélisable au niveau du trafic réseau qu'au niveau d'un utilisateur ou d'un système d'exploitation. Le développement de cet outil en interne reste encore actif, car il n'a pas encore atteint un niveau de maturité suffisant dans l'automatisation des alertes et dans son autonomie d'analyse. Pour atteindre ces objectifs, l'analyse de flux a cependant démontré un avantage précieux par rapport à celui de l'analyse de paquets : elle est rejouable. Le simple fait de conserver les flux, alors qu'on ne conserve pas les paquets, permet de tester ou de valider un nouveau scénario sur des flux du semestre dernier, rendant par là même possible la découverte d'un incident potentiel avec quelques mois de retard, mais mieux vaut tard que jamais...

Références

[1] Description du format propriétaire NetFlow, **http://en.wikipedia.org/wiki/Netflow**

[2] NetSA (SiLK, YAF, etc.), **http://tools.netsa.cert.org/**

[3] MaxMind LLC – Outils de géolocalisation et de détection de fraude en ligne, **http://www.maxmind.com/**

[4] Phoenix Labs PeerGuardian, **http://phoenixlabs.org/pg2/**

[5] IP4R, extension PostgreSQL pour la gestion des adresses et réseaux IPv4, **http://pgfoundry.org/projects/ip4r/**

[6] FISCHBACH (Nicolas), « Les flux réseau », MISC n°17, pages 72-75.

[7] *Argus Open Project : the network Audit Record Generation and Utilization System*, **http://qosient.com/argus/**

[8] OSSIM : *Open Source Security Information Management*, **http://www.ossim.net/**

[9] NTOP : *network traffic probe that shows the network usage*, http://www.ntop.org/

[10] Draft IETF sur le support biflow dans IPFIX, **http://www.ietf.org/internet-drafts/draft-ietf-ipfix-biflow-05.txt**

[11] LAU (Stephen), « *The Spinning Cube of Potential Doom* », décembre 2003, **http://www.nersc.gov/nusers/security/TheSpinningCube.php**

[12] PostGIS, extension PostgreSQL en base de données spatiale, **http://postgis.refractions.net/**

[13] WorldKit, application flash pour la cartographie en ligne, **http://worldkit.org/**

Reverse proxy Apache 2.2 : le couteau suisse sécurité de vos applications

La plupart des directions informatiques en entreprise ont été confrontées à la problématique de la mise à disposition d'applications internes pour leurs utilisateurs connectés sur l'internet et non au sein de leur réseau local. Au travers de cette fiche technique, nous allons voir comment Apache 2.2 peut vous aider à fournir ce service avec un minimum de sécurité. Rien n'est parfait bien sûr, mais cela vaut la peine d'essayer.

mots clés : *reverse proxy / sécurisation transparente pour l'application / Apache / ssl / ldap*

1. Besoins de d'ouverture, besoins de sécurité

1.1 Le contexte

L'accès depuis l'extérieur du réseau doit être possible pour un certain nombre d'applications afin de permettre aux membres de l'entreprise de travailler de manière fluide et simple : commerciaux en clientèle, techniciens en intervention sur des sites clients, consultants en mission, etc.

1.2 Traitons de la mise en ligne d'une boîte noire

Le cas sur lequel nous allons nous concentrer est celui des applications de type « boite noire » pour lesquelles vous n'avez pas les sources : progiciels, applications métiers commerciales ou toutes sortes d'appliances techniques. En effet, dans ce cas, un contrôle externe comme celui fourni par un reverse proxy est la seule solution pour éviter un tant soit peu de mettre en ligne sur le net une bombe à retardement pour votre réseau interne.

2. Reverse proxy : pourquoi et comment ?

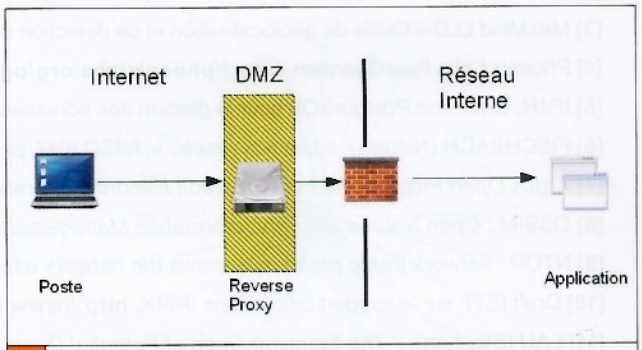
2.1 Comprenons d'abord les risques

Une application interne est souvent vue comme évoluant dans un domaine de confiance, le réseau interne. La **confidentialité des données (authentification, données applicatives, etc.)** transitant entre le client et le serveur est souvent considérée comme assurée de facto. Le serveur applicatif interne peut être totalement dépourvu de couche technique permettant d'implémenter cette confidentialité. Dans le cas d'une mise en ligne sur l'internet, cette confidentialité devient alors une véritable problématique.

Un second risque, toujours issu d'une excessive confiance dans le réseau interne, est celui de voir l'application dotée d'une couche d'**authentification et d'habilitation** basique, voire inexistante. Le *reverse proxy* a alors l'obligation de fournir ces fonctions.

Un troisième risque est celui de la **faille applicative**. En effet, mettre en ligne une application sur le net augmente de manière significative son exposition aux attaques en tous genres. Le cas particulier de la boîte noire nous oblige à travailler en amont de l'accès avec un équipement filtrant comme un reverse proxy.

2.2 Reverse proxy : rôle et positionnement



1 *Accès depuis Internet à l'application via un reverse proxy*

Ce reverse proxy `http/https` aura pour mission de, autant que faire se peut, réduire les risques d'intrusion décrits précédemment. Pour cela, il doit :

- fournir une communication chiffrée entre le navigateur du collaborateur sur Internet et le relais en DMZ ;
- rajouter si besoin une authentification supplémentaire à l'application ;
- filtrer les données envoyées par le navigateur à l'application.

Sur la figure 1, l'application est dans le réseau interne. Cependant, il s'agit d'une vue logique. En effet, afin d'augmenter la sécurité, l'application peut très bien être implantée dans une DMZ l'isolant tant de la DMZ publique que du réseau interne. Rien n'empêche non plus de mettre côté interne un reverse proxy en frontal de l'application afin de se doter aussi de capacité de filtrage en interne.

Le reverse proxy que nous allons utiliser ici est **Apache en version 2.2**. La version 2.2 apporte entre autres choses des capacités d'authentification étendues dont nous explorerons une partie dans le chapitre 4.3. De même, elle amène aussi des fonctions de répartitions de charge permettant d'adresser de vrais besoins d'industrialisation de la mise en production d'applications, mais que l'on ne traitera pas ici, car ne touchant pas directement à la sécurité.

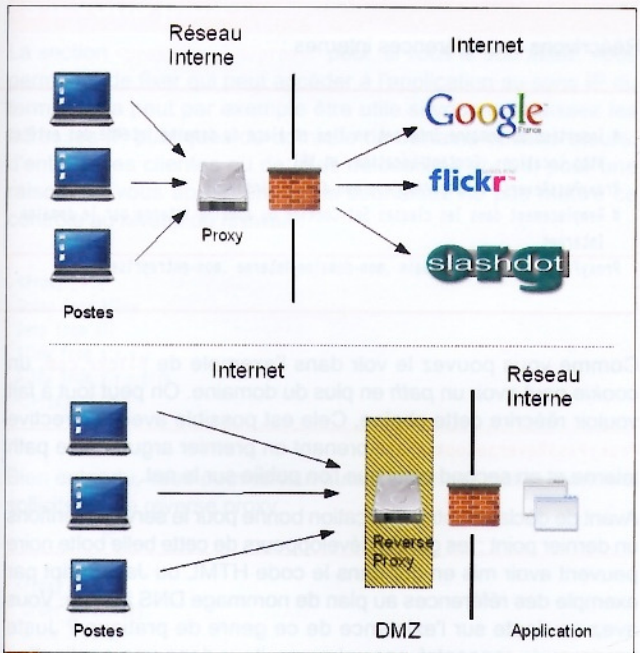
Christophe Brocas
christophe.brocas@free.fr

encadré 1

Proxy VS Reverse Proxy : éclairage

Un proxy HTTP relaie les requêtes d'N utilisateurs vers les serveurs web qu'ils souhaitent atteindre. Grâce à ce rôle de relais, il peut fournir des fonctions de cache en fournissant une copie locale des pages déjà accédées lors des accès ultérieurs améliorant ainsi la vitesse de consultation, des fonctions de contrôle d'accès aux sites web (listes blanches ou noires de sites consultables) ou de filtrage des ressources rapatriées (analyse antivirale, analyse des scripts dans les pages, etc.). Les logs d'un proxy HTTP permettent de lister les tentatives d'accès à des sites interdits, de consolider une liste des sites les plus accédés, des utilisateurs les plus gourmands en bande passante, etc.

Un reverse proxy HTTP est, quant à lui, un composant localisé du côté du serveur que le client cherche à accéder. Le reverse proxy est le serveur qui va dialoguer avec le client en lieu et place du serveur vers lequel il relaie les requêtes des utilisateurs. Il assure des fonctions de cache en ne relayant pas un certain nombre de requêtes vers le serveur applicatif, de terminaisons SSL allégeant ici aussi la charge du serveur, mais aussi de sécurité (contrôle d'accès via une habilitation LDAP par exemple, analyse du contenu des paramètres des requêtes HTTP POST ou GET, conformité des échanges HTTP). Les logs d'un reverse proxy permettent de connaître les clients sollicitant le serveur applicatif, de lister les URL accédées sur le serveur, mais aussi le contenu des paramètres (partie DATA des requêtes HTTP), etc. Ce sont ces aspects qui vont nous intéresser.



2 *Rôle et positionnement d'un proxy et d'un reverse proxy*

3. Mon reverse proxy, petit à petit...

3.1 Avant tout, quelques pré-requis

Première précision : l'architecture qui suit a été testée et validée sur une distribution Ubuntu 7.04. Le reverse proxy était un Apache 2.2.4 [1] adossé à OpenSSL 0.9.8 et au module `mod_proxy_html` 2.5.2 [2]. Nous ne détaillerons pas le processus d'installation des produits : vous y allez avec vos petits doigts à grands coups de `sudo apt-get install apache2 openssh...` et tout devrait rouler :-) Apache, OpenSSL et le module `mod_proxy_html` installés, il faudra activer les modules Apache suivants (s'ils ne le sont pas déjà) via la commande `a2enmod nom-module` :

```
mod_proxy,
mod_proxy_http,
mod_proxy_connect,
mod_proxy_html (2.5 ou >),
mod_ssl,
mod_auth_basic,
mod_authz_default,
mod_authz_host,
mod_authnz_ldap,
mod_ldap.
```

3.2 Et avant de le sécuriser, si on le faisait marcher cet accès ?

C'est en effet une remarque toute bête, mais avant de penser à la sécurité, il faut se préoccuper de mettre en œuvre correctement l'accès externe à cette application interne.

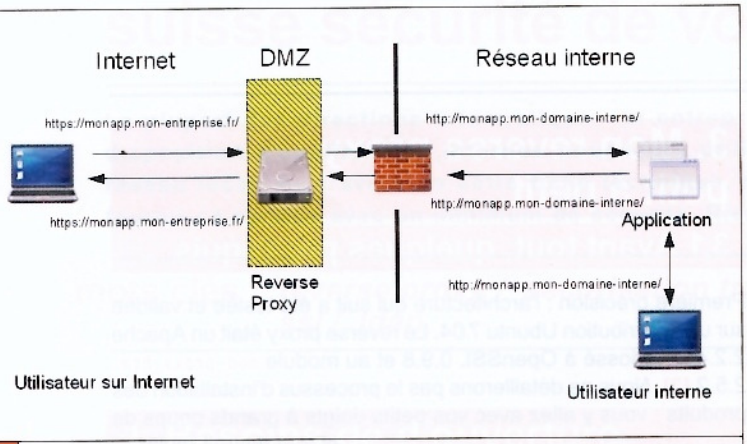
La situation est la suivante :

- On a une application interne qui est accédée au sein du réseau interne via l'URL suivante : **http://monapp.mon-domaine-interne/**.
- On va l'accéder depuis l'Internet via l'URL : **https://monapp.mon-entreprise.fr/**.

Tout d'abord, comprenons ce qui est en jeu dans cet accès :

- Au sein du réseau interne, lors des accès à l'application via l'URL interne, toutes les requêtes HTTP ainsi que tous les champs d'en-têtes de requêtes HTTP comme les champs `Location:` ou `Set-Cookie:` font référence au nom DNS interne de l'application.
- Le contenu même des pages générées par le serveur peut utiliser des références au nom DNS interne de l'application.

Voyons donc comment gérer ces situations afin de ne pas avoir d'effet de bord lié à l'URL utilisée pour solliciter l'application.



3 *Plan de nommage et rôle du reverse proxy*

3.3 Le fichier de conf'

Le fichier de configuration (car il n'y en a qu'un !) est là : `mods-enabled/proxy.conf`. Sa structure est la suivante :

```
<IfModule mod_proxy.c>

Directives générales concernant le paramétrage du proxy.

<Proxy *>
Directives contrôlant l'accès au reverse proxy.
Ce paragraphe accueillera aussi les fonctions d'authentification.
</Proxy>

<VirtualHost adresse:port>
Toutes les directives de réécritures seront ici.
</VirtualHost>

</IfModule>
```

3.4 On a dit : « Pas de relais ici ! »

Tout d'abord, inactivons la fonction de proxy ouvert via la directive suivante :

```
ProxyRequests off
```

En effet, par défaut, le proxy est ouvert aux requêtes d'où qu'elles viennent. Cette directive évite que nous ayons un vrai relais en bonne et due forme permettant à n'importe qui d'accéder à n'importe quelle ressource sur Internet via votre machine.

3.5 Accueillons nos hôtes ;-)

Comme nous l'avons dit au point 3.2, nous allons choisir d'accueillir nos utilisateurs sur le nom DNS public `monapp.mon-entreprise.fr`.

Choisissons donc l'adresse IP publique (et le port) sur laquelle nous réceptionnons les requêtes des utilisateurs :

```
Listen 443
NameVirtualHost x.x.x.x:443
```

Ce choix n'est pas une obligation et on peut accueillir ces mêmes requêtes sur toutes les interfaces de la machine via `NameVirtualHost *:443`.

Ensuite, adaptons le paragraphe `VirtualHost` qui doit être en phase avec la directive `NameVirtualHost` :

```
<VirtualHost x.x.x.x:443>
 ServerName monapp.mon-entreprise.fr
 [...]
</VirtualHost>
```

3.6 Et si nous la transformions en application Internet, notre application interne !

Comme nous l'avons vu au chapitre 3.2, les utilisateurs Internet sollicitent l'adresse `monapp.mon-entreprise.fr`. Commençons donc par réécrire ces requêtes arrivant sur l'adresse **https://monapp.mon-entreprise.fr/** en requêtes sollicitant l'adresse **http://monapp.mon-domaine-interne/** :

```
# Redirection de l'adresse Internet vers l'adresse interne
ProxyPass        / http://monapp.mon-domaine-interne/
```

Ensuite, le serveur d'application renvoie des réponses HTTP avec des en-têtes de type `Location:` (par ex. lors des redirections sur les réponses HTTP de code 301) ou `Set-Cookie:`. Or, ces en-têtes peuvent contenir des références à l'adressage interne. Exemple de `Set-Cookie:` sur le site de partage de photos Flickr :

```
Set-Cookie: cookie_session=2532027%3Ab5209e62dcb4341fa5c8221852221091;
          path=/; domain=flickr.com
```

Réécrivons ces références internes :

```
# Insertion du domaine Internet en lieu et place du domaine interne des entêtes
  http Location:, Content-Location: et URI:
ProxyPassReverse / http://monapp.mon-domaine-interne/
# Remplacement dans les clauses Set-Cookie: du domaine interne par le domaine
  Internet
ProxyPassReverseCookieDomain .mon-domaine-interne .mon-entreprise.fr
```

Comme vous pouvez le voir dans l'exemple de `flickr.com`, un *cookie* peut avoir un *path* en plus du domaine. On peut tout à fait vouloir réécrire cette chaîne. Cela est possible avec la directive `ProxyPassReverseCookiePath` prenant en premier argument le path interne et en second celui que l'on publie sur le net.

Avant de déclarer votre application bonne pour le service, vérifions un dernier point : les gentils développeurs de cette belle boîte noire peuvent avoir mis en dur dans le code HTML ou JavaScript par exemple des références au plan de nommage DNS interne. Vous avez un doute sur l'existence de ce genre de pratique ? Juste un exemple rencontré par votre serviteur dans une application commerciale actuelle :

```
<HTML>
<HEAD>
<script>
[...]
var BaseURL = '' + 'http://url interne/';
</script>.
```

Pour gérer ce genre de situation, il faut faire appel au module non fourni par Apache, `mod_proxy_html`. Celui-ci scrute le contenu des pages et permet les substitutions au sein de la partie DATA des réponses HTTP. Voici les directives à utiliser :

```
# Activation du module mod_proxy_html
SetOutputFilter proxy-html
ProxyHTMLExtended On

# Remplacement de l'adresse interne par l'adresse publique
ProxyHTMLURLMap http://monapp.mon-domaine-interne https://monapp.mon-
                entreprise.fr
```

Attention

Ces remplacements sont dépendants de la forme que prennent ces adresses au sein des pages.

Et voilà, notre application est désormais prête à être sollicitée depuis l'Internet.

4. Bon ok, on sécurise, on sécurise...

4.1 Bien blanche, la liste, s'il vous plait !

La section `<proxy> ... </proxy>` peut, si vous le souhaitez, vous permettre de fixer qui peut accéder à l'application au sens IP du terme. Cela peut par exemple être utile si vous connaissez les adresses IP publiques du domicile de certains collaborateurs, d'entreprises clientes ou de sites délocalisés et que, pour une raison qui vous appartient, vous souhaitiez ne pas mettre ce contrôle au niveau du *firewall* :

```
<Proxy *>
Order Deny,Allow
Deny from all
Allow from x.x.x.x
Allow from y.y.y.0/24
</Proxy>
```

Bien entendu, vous pouvez tout autant autoriser tout le monde à solliciter votre reverse proxy :

```
<Proxy *>
 Order allow,deny
 Allow from all
</Proxy>
```

4.2 No sniffer please !

Afin de préserver la confidentialité entre le client et le serveur, il faut pouvoir accéder à l'application en mode sécurisé. Pour cela, nous allons mettre en place un canal de communication chiffré et utiliser un certificat. Le certificat doit avoir comme valeur *Common Name* (ou CN) le nom `monapp.mon-entreprise.fr`. Ce certificat permet d'assurer à l'utilisateur sur Internet qui cherche à joindre le serveur `monapp.mon-entreprise.fr` qu'il est bien en contact avec le bon serveur.

Cela se fait en activant le moteur SSL d'Apache et indiquant où trouver le certificat et la clé privée. Voici les directives à ajouter à la section VirtualHost.

```
# activation du moteur SSL
SSLEngine on

# Localisation du certificat et de la clé privée
SSLCertificateFile    /etc/apache2/server.crt
SSLCertificateKeyFile /etc/apache2/server.key
```

4.3 LDAP : identifions, habilitons

L'un des intérêts du composant reverse proxy est de pouvoir permettre de rajouter des fonctionnalités à une application. L'une de ces fonctionnalités est le rajout d'une phase d'authentification/habilitation si l'application n'en possède pas en natif ou si l'on désire en rajouter une supplémentaire.

La section <proxy> accueille deux fonctions : l'authentification d'abord et ensuite l'habilitation. Cette première section qui suit fixe la méthode d'authentification, ici LDAP. On fournit la localisation du serveur LDAP, le base DN, le champ servant à identifier l'utilisateur (ici `uid`) ainsi que le compte utilisé pour faire le contrôle LDAP.

```
# Configuration de l'authentification
AuthName "Authentification LDAP"
AuthBasicProvider ldap
AuthType Basic

# Adresse de l'annuaire LDAP
AuthLDAPURL ldap://x.x.x.x:Port/dc=<mon org>,dc=fr?uid?sub?(objectClass=*)

# User/pwd du compte qui permet de verifier l'authentification et l'habilitation du client
AuthLDAPBindDN <DN du compte>
AuthLDAPBindPassword <password>
```

Une fois que l'utilisateur est identifié, on peut contrôler s'il est habilité à utiliser l'application. Plusieurs moyens sont utilisables en LDAP pour cela. Illustrons en deux : le premier extrait de code contrôle si l'UID fourni par le client identifie une entrée d'annuaire ayant le bon attribut avec la bonne valeur :

```
require ldap-attribute <nom de l'attribut qui stocke les habilitations>=<valeur
                        fixant un droit sur l'application>
```

et l'extrait qui suit qui contrôle si l'UID fourni par le client est membre d'un groupe LDAP (par défaut, le DN de cet UID doit figurer dans une clause `uniquemember` du groupe mis en paramètre de la clause ci-dessous) :

```
require ldap-group <DN du groupe LDAP>
```

4.4 Filtrage des données émises vers l'application

`ModSecurity` **[3]** est un module Apache se comportant comme un pare-feu applicatif : analyse des requêtes/réponses HTTP, transformation de flux HTTP, logs détaillés. Je ne reviendrai pas sur l'utilisation de `ModSecurity` déjà parfaitement décrite par les auteurs de NuFW dans le numéro 20 de MISC **[4]**. Je n'utilise l'exemple de `ModSecurity` que pour illustrer l'intérêt d'avoir un reverse proxy permettant de filtrer les données émises vers l'application par l'utilisateur.

Imaginons la situation suivante :

- Un site institutionnel à forte visibilité se retrouve hacké.
- L'analyse des traces de connexions permet de trouver des formulaires comportant des injections SQL.
- L'analyse de l'injection permet de s'apercevoir que ces injections touchent un script en particulier et se concentrent sur le paramètre `uniqid` (ex. : saisie de côtes et de caractères alphabétiques en lieu et place de 1 à 3 caractères numériques).
- Et là, si votre application est accédée en direct et sans coupure depuis l'Internet... **c'est le drame !**

Les raisons de cette invocation volontairement dramatisée (quoique !) sont les suivantes. Sans reverse proxy, vous devez :

- Arrêter la boîte noire et donc le site !
- Solliciter le support éditeur en priant pour que leur support puisse vous fournir un patch rapide et efficace pour ce formulaire.

Le souci dans tout cela, c'est que le site est à l'arrêt ! Or, au vu du diagnostic initial, la meilleure réponse en termes d'*uptime* du site est de coupler les actions ci-dessus avec des règles `ModSecurity` au niveau Reverse Proxy qui force certains contrôles sur le paramètre du formulaire en question.

Nous obtenons les gains suivants :

- l'ouverture du site ;
- la sécurisation de la faille par filtrage ;
- et la demande d'évolution structurelle de l'application/progiciel qui est traitée en mode asynchrone par les développeurs de l'éditeur.

Le patch `ModSecurity` par contrôle de la valeur numérique sur 5 positions du paramètre `uniqid` se fait par rajout des directives suivantes dans la section VirtualHost :

```
SecAuditEngine RelevantOnly
SecRuleEngine  On
SecRule ARGS:uniqid "!^\d{1,5}$"
SecDefaultAction log,auditlog,deny,status:403,phase:2,t:none
```

Conclusion

Nous venons de voir comment une architecture à base de reverse proxy donne une grande souplesse aux ingénieurs réseau et sécurité en termes de mises à disposition d'applications sur Internet. Si l'on doit ne retenir qu'un avantage parmi ceux que procurent un reverse proxy, c'est sa capacité à fournir de nombreux services à valeur ajoutée de manière transparente de l'application. Ingénieurs sécurité et réseau : vos directions utilisatrices, vos équipes de développement et vos équipes de veille sécurité vous remercieront sûrement d'utiliser un reverse proxy !

Crédits

Je souhaite remercier Sarah Nataf, ainsi que Jean-Michel Farin pour leur relecture impliquée et pertinente qui ont permis d'améliorer la structure de cet article, ainsi que sa cohérence d'ensemble et sa précision.

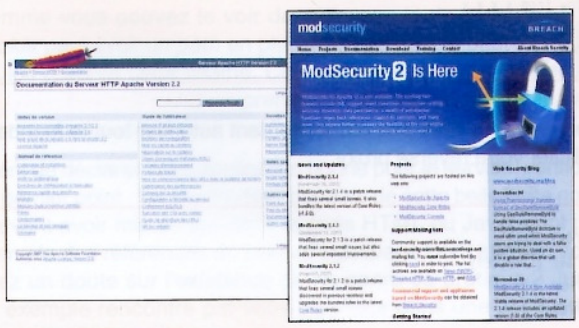
Références

[1] Serveur web Apache 2.2, **http://httpd.apache.org/docs/2.2/fr/**

[2] mod_proxy_html, **http://apache.webthing.com/mod_proxy_html/**

[3] modsecurity, **http://www.modsecurity.org/**

[4] DEFFONTAINES (Vincent) et LEBLOND (Éric), « Sécurité avancée du serveur web Apache : mod_security et mod_dosevasive », *MISC* n°20 juillet/août 2005.



La sécurité des communications vocales (1) : le codage de la voix

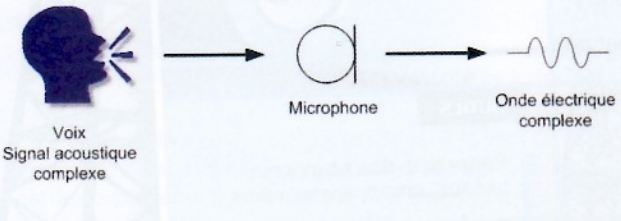
Le codage et la transmission de la parole sont deux éléments essentiels dans les télécommunications modernes : si la part du trafic des données prend une importance grandissante depuis plusieurs années, les communications vocales restent très importantes non seulement d'un point de vue quantitatif, mais également d'un point de vue qualitatif et « émotionnel ». Contrairement aux données, un message vocal est infiniment plus riche et contient à la fois les données propres au message, mais également aux acteurs de la transmission. C'est précisément la raison pour laquelle le codage, la transmission et la sécurisation de la voix relèvent de techniques beaucoup plus complexes que ceux des données. Dans cet article, le premier d'une série de trois, nous expliquons comment se fait le codage et le traitement de la voix. Il est essentiel de bien comprendre la structure d'un signal sonore, les techniques permettant de le représenter et de le traiter si l'on veut être capable d'en assurer la sécurité, non seulement en termes de confidentialité, mais également en termes d'intégrité et d'authentification, lesquels seront présentés dans les deux articles à suivre.

mots clés : ***communication vocale / voix / analyse de Fourier / signal analogique / sinusoïde***

1. Introduction

Deux types de signaux peuvent être considérés dans le codage et la transmission de la parole :

- Les signaux digitaux (ou numériques) constitués d'un nombre fini d'états discrets (dans le cas du binaire 0 et 1). Ces signaux sont typiquement ceux produits par un ordinateur (ou par l'humain assisté par un ordinateur via un clavier par exemple).
- Les signaux analogiques, qui sont en fait une variation continue et dynamique sur un intervalle de définition beaucoup plus riche (en général l'ensemble des réels). Ces signaux, d'une extrême richesse, sont typiquement produits par l'homme (et plus généralement les êtres vivants) et lui sont spécifiques.



1 *Codage de la voix*

De fait, la communication de la parole est la plus importante à bien des niveaux, et en particulier du fait de son extrême richesse et sa capacité à transmettre plusieurs niveaux de messages : la voix contient non seulement des informations proprement dites (le message), mais également des informations périphériques concernant son auteur (la couleur du message). Dans toute communication, il est donc nécessaire de traiter à la fois le message et sa coloration, problème qui n'existe pas pour la communication des données.

En reprenant les trois problèmes identifiés par C. Shannon [1] dans toute communication, le traitement des communications vocales se doit de considérer les aspects suivants :

- Comment représenter (ou coder) la parole (théorème de codage de source).
- Comment lutter contre les effets perturbateurs de la nature contre le canal (théorème de codage d'erreur et d'information mutuelle).
- Comment protéger un message vocal contre des écoutes illégitimes (théorème du secret parfait).

Concernant la qualité de la transmission, il est essentiel de préciser que quel que soit le système ou la technique de traitement de la parole utilisés, ces derniers doivent dégrader le moins possible le signal vocal original – il y a en effet toujours une réduction de qualité plus ou moins grande lors du passage voix/microphone. De plus, tout procédé supplémentaire visant à sécuriser un signal vocal réduira inévitablement encore plus la qualité sonore du signal. Il est donc nécessaire de considérer le couple canal de communication/système de protection pour aboutir finalement au meilleur compromis en termes de qualité sonore. Considérer l'un sans l'autre n'a pas de sens. Il est donc essentiel de bien comprendre, afin de savoir comment sécuriser la voix de manière industriellement acceptable (qualité du service), comment on la code et on la traite.

Concernant spécifiquement la sécurisation de la voix contre une écoute illégitime, la question essentielle est la suivante : quelle intelligibilité résiduelle subsiste-t-il dans un signal vocal protégé (typiquement brouillé) ? Mais il est essentiel de conserver à l'esprit que ce souci de sécurité est plus complexe à gérer dans le cas spécifique de la voix. L'attaquant ne doit accéder ni au message, ni à sa coloration. Ainsi, déterminer qui parle – si cette information n'est pas protégée – peut être aussi important que le message lui-même.

La sécurité des communications vocales sera traitée à travers trois articles. Cet article explique comment est représentée et traitée la voix. Le second article [2] présentera les techniques permettant de protéger les communications vocales dans le cas analogique. Si les techniques (analogique →) numériques sont de nos jours les plus utilisées, notamment dans les systèmes commerciaux, les techniques purement analogiques restent d'un grand intérêt pour certains systèmes opérationnels, bien que paradoxalement elles

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

restent assez méconnues. Un signal analogique brouillé reste un signal continu dont la richesse et la complexité mathématique (plus grande que celle des signaux numériques) constituent un obstacle « plus important » pour les outils d'analyse actuels. Cependant, là où de tels outils peuvent échouer, une oreille humaine exercée ou aiguë pourra saisir des fragments de communication s'il subsiste une intelligibilité résiduelle dans le signal brouillé. Le troisième et dernier article [3] exposera en détail les techniques de protection des signaux vocaux numériques, illustrées par l'analyse d'un tel système, utilisé par l'armée de l'air chinoise.

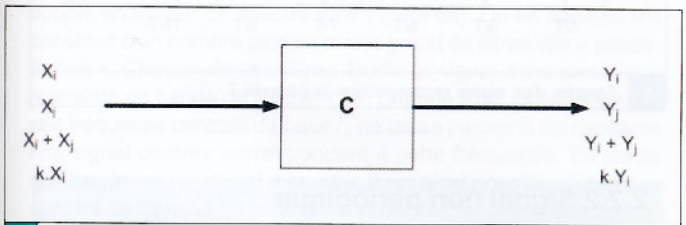
Dans cet article, afin de ne pas alourdir le propos, nous ne présenterons pas les aspects purement électroniques du traitement de la parole. Le lecteur pourra consulter [4].

2. La voix et son traitement

Nous allons dans cette partie voir quels sont les outils mathématiques et les dispositifs électroniques de base utilisés pour traiter les ondes électroniques complexes formant un signal vocal (Figure 1). Cette compréhension est fondamentale pour saisir comment il est ensuite possible de sécuriser un tel signal.

2.1 Concepts électroniques et mathématiques de base

Nous considérerons un circuit (ou composant) électronique générique C recevant un signal d'entrée X_i, le traitant et produisant un signal de sortie Y_i (Figure 2).



2 *Circuit linéaire*

Le circuit C est donc tel que $Y_i = C(X_i)$. Il réalise un traitement sur X_i (que l'on peut représenter par une fonction mathématique plus ou moins complexe, qu'il ne sera pas utile d'expliciter). Un circuit linéaire C doit satisfaire deux propriétés :

- la propriété d'homogénéité telle que $C(k\,X_i) = k\,C(X_i) = k\,Y_i$;
- la propriété de superposition telle que $C(X_i + X_j) = C(X_i) + C(X_j) = Y_i + Y_j$.

Ces deux propriétés sont fondamentales pour effectuer les opérations de base utilisées dans le traitement de la parole = la superposition de signaux et l'amplification de signal. La plupart des composants électroniques employés dans le traitement de la parole sont linéaires ou, du moins, le sont dans un intervalle suffisant donné (pour les valeurs de X_i et de Y_i), appelé « intervalle dynamique ». L'intérêt de ces circuits réside dans la propriété de linéarité de la fonction associée [N1]. En effet, il est impossible de considérer tous les signaux d'entrée X_i existants. Mais, des résultats d'algèbre linéaire basique [5] permettent de considérer seulement un sous-ensemble réduit de signaux caractéristiques, lesquels permettent ensuite par combinaison linéaire de reconstituer tous les signaux existants. Ce sous-ensemble est appelé l'ensemble générateur de base des signaux $B = (X_1, X_2, \ldots, X_n)$. Ainsi, pour tout signal X, on peut l'écrire sous la forme

$$X = k_1.X_1 + k_2.X_2 + \ldots + k_n.X_n$$

où les valeurs k_i sont des valeurs réelles.

Alors, tout signal complexe X (et le signal résultat Y produit par C) peut être aisément défini par un nombre réduit de signaux caractéristiques. Il suffit de connaître pour chacun des X_i de l'ensemble B, le résultat Y_i par C. Ce qui donne :

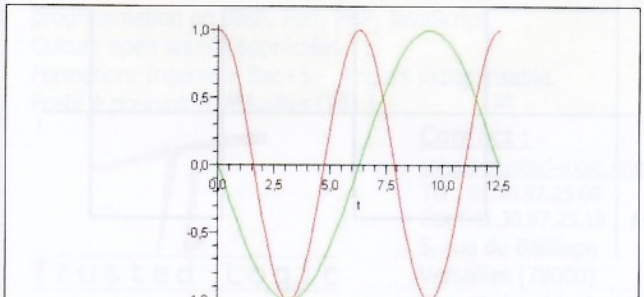
$$Y = C(X) = k_1.Y_1 + k_2.Y_2 + \ldots + k_n.Y_n$$

Il reste maintenant à trouver de tels ensembles B. Sans entrer dans les détails, les signaux de base les plus utilisés (parce que mathématiquement plus pratiques à manipuler) sont les signaux sinusoïdaux (ou sinusoïdes). Un tel signal est défini par la fonction :

$A\cos(\omega t + \varphi)$ ou $A\sin(\omega t + \varphi)$

où

- t représente le temps.
- A est l'amplitude du signal (valeur maximale du signal lorsque la fonction cosinus ou sinus valent 1).
- ω est la fréquence angulaire ; pour expliciter ce paramètre, rappelons que les fonctions sinus et cosinus sont périodiques, de période 2π [N2], et donc les fonctions sinusoïdes. La période pouvant, pour ces dernières, être plus petite ou plus grande, on note la période $T = 2\pi/\omega$. Il est cependant plus pratique (étant les ordres de grandeur des périodes) de considérer le nombre de périodes par unité de temps (la seconde), autrement dit la fréquence $f = 1/T = \omega/2\pi$, mesurée en Hertz (Hz). Le terme de fréquence angulaire vient alors du fait que $\omega = 2\pi.f$. Notons que c'est la fréquence qui détermine si un signal est grave (fréquence basse) ou aigu (fréquence élevée).
- φ est la phase, autrement dit la valeur du signal lorsque $t = 0$. Le terme de déphasage se rapporte à deux signaux n'ayant pas la même phase (la valeur $\varphi_2 - \varphi_1$ est alors la différence de phase).



3 *Sinusoïdes (en rouge cos(t) et en vert cos(0.5t + π/2))*

Notons que l'amplification d'un signal (multiplier par k) ne modifie ni la fréquence (un signal grave reste grave même une fois amplifié), ni la phase.

Maintenant que les briques et les outils de base sont définis, voyons comment en pratique tout signal complexe se décompose. Pour cela, on utilise la théorie mathématique appelée « analyse de Fourier » [6].

2.2 Analyse de Fourier du signal

Deux cas sont à considérer selon la nature du signal étudié.

2.2.1 Signal naturellement périodique

Ces signaux sont donc représentables par des fonctions périodiques. L'analyse de Fourier, dont l'outil central est la série de Fourier, consiste à représenter un tel signal comme une somme (en fait du point de vue de l'algèbre linéaire comme une combinaison linéaire ; ou du point de vue de l'électronique comme une superposition de signaux plus ou moins amplifiés) plus ou moins complexe de sinusoïdes.

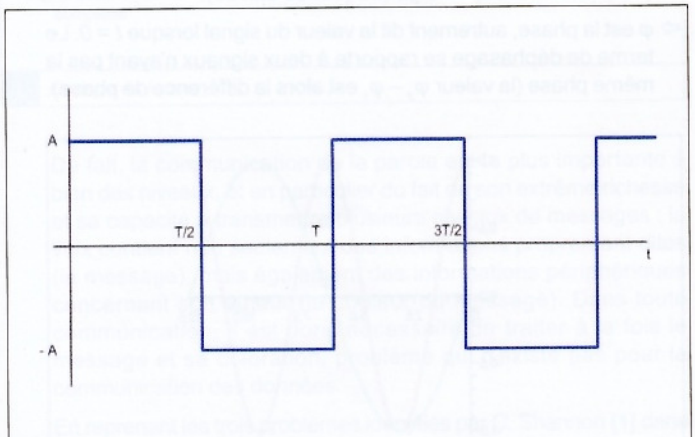
Soit f(t) un signal périodique. On peut donc l'écrire de la manière suivante :

(1)

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(n.\omega_1 t) + \sum_{n=1}^{\infty} b_n \sin(n.\omega_1 t)$$

avec la convention que $\omega_1 = 2\pi f_1$. La fréquence du signal est donnée par la valeur f_1 qui représente donc la fréquence fondamentale. Les autres termes en cosinus et sinus représentent les fréquences dites « harmoniques » $n.\omega_1$ de la fréquence angulaire fondamentale ω_1. Ce sont notamment les harmoniques qui déterminent la coloration d'un signal vocal. L'ensemble constitué de la fréquence fondamentale et des fréquences harmoniques s'appelle le « spectre du signal ». Notons que les coefficients a_n et b_n se calculent en intégrant la fonction $f(t)$ sur un cycle complet (de longueur $1/f_1$).

Afin d'illustrer le propos, considérons le signal rectangulaire suivant de période T (figure 4).



4 *Signal rectangulaire de période T*

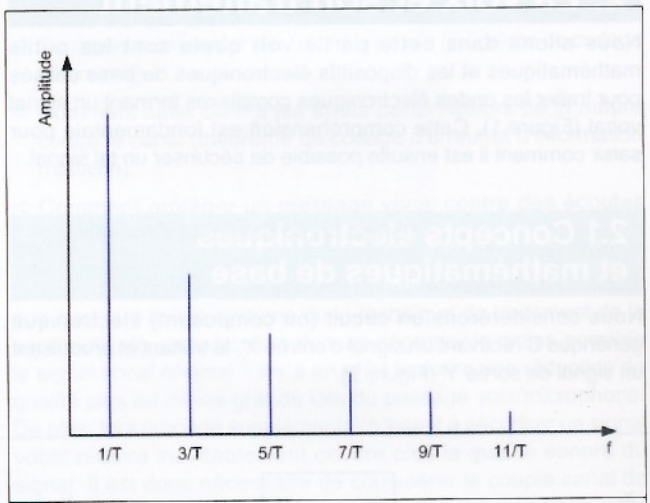
Il est défini par la fonction $f(t)$ suivante avec $\omega_1 = 2\pi/T$:

$$f(t) = \begin{cases} A & si \quad 0 \le t \le \dfrac{T}{2} \\ -A & autrement \end{cases}$$

En utilisant l'analyse de Fourier, cette fonction s'écrit de manière équivalente par :

$$f(t) = \sum_{n=1 et\, n\, impair}^{\infty} \frac{4A}{n.\pi} \sin(n.\omega_1 t)$$

Son spectre est alors représenté par la Figure 5.



5 *Spectre d'un signal rectangulaire de période T*

2.2.2 Signal non périodique

Dans le cas d'un signal non périodique (ou apériodique) – typiquement la parole –, le modèle précédent se complique quelque peu. En fait, la notion classique de série de Fourier ne s'applique plus, du moins pas directement. Intuitivement, un signal apériodique peut être vu comme un signal de période à une seule période, correspondant à la longueur du signal lui-même (le signal ne se répète jamais en totalité). Autrement dit, $T \to +\infty$ ou de manière équivalente $f \to 0$. La fréquence du signal est infinitésimalement petite.

Les fréquences harmoniques (multiples de la fréquence fondamentale) prennent alors n'importe quelles valeurs (passage du discret au continu) ce qui nécessite de considérer dans la formule (1), non plus des sommations, mais des intégrations. Afin de faciliter les calculs, il nécessaire également de considérer une formulation quelque peu différente de la série de Fourier.

On va utiliser le fait que toute somme de la forme $a.cos(x) + b.sin(x)$ peut s'écrire sous la forme $k.cos(x + y)$ où k et y dépendent

de *a* et *b* (calculs simples de trigonométrie). L'équation (1) utilise alors des termes de la forme $k_n.cos(n.\omega_1 + \varphi_n)$ où k_n et φ_n sont des paramètres qui dépendent de a_n et b_n.

Le passage au continu (cas d'un signal apériodique) donne la formule suivante **[N3]** :

$$f(t) = \frac{1}{\pi}\int_{1}^{\infty} k(\omega).\cos(\omega t + \varphi(\omega))d\omega$$

Pour caractériser alors totalement le signal, il faut déterminer les fonctions $k(\omega)$ et $\varphi(\omega)$. Cependant, contrairement au cas périodique, les outils mathématiques deviennent trop complexes pour déterminer ces fonctions. La physique prend alors le relais et exploite le fait qu'un signal est essentiellement déterminé par une intensité, une tension et une puissance, lesquelles peuvent être mesurés à tout instant t :

- La tension exprimée en volts (*V*).
- L'intensité, exprimée en ampères (*A*).
- La puissance, exprimée en watts (*W*). Cette dernière quantité caractérise le taux de variation de l'énergie du signal. Si la tension du signal est de V volts, son intensité est de I ampères alors sa puissance *P* est donnée par *P = V.I*. En faisant intervenir la notion de résistance *R* du circuit, exprimée en ohms, la loi d'Ohm, selon laquelle nous avons *V = R.I*, implique que *P* est proportionnel à V^2. Il est ainsi beaucoup plus pratique de considérer cette dernière valeur pour étudier un signal. Elle permet d'établir le *spectre de densité de puissance*, équivalent au spectre de fréquences d'un signal périodique.

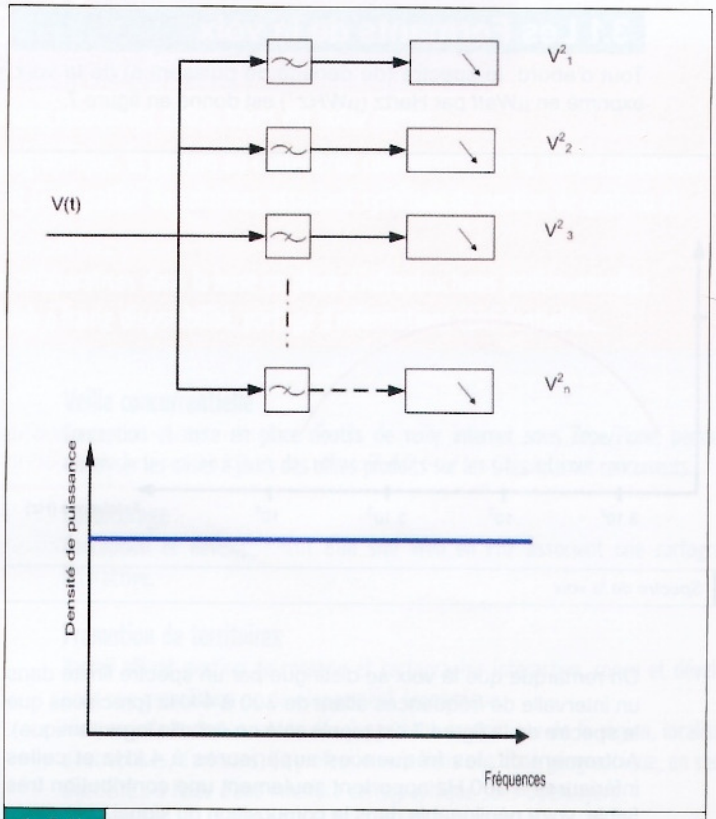
Ce spectre est établi de manière simple en utilisant un appareil appelé analyseur de spectre (voir Figure 6a). Un tel appareil est constitué d'un nombre plus ou moins grand de filtres dits « passe-bande ». Chacun de ces filtres étudie le signal dans une plage restreinte de bande de fréquences. Chaque filtre, caractérisé par une fréquence centrale de base f_i, ne laisse passer la composante i du signal centrée correspondant à cette fréquence. En sortie de filtre, la valeur V^2_i est mesurée. Il est ainsi possible d'établir le spectre de densité de puissance du signal, et ainsi caractériser entièrement ce dernier. Il est évident que plus le nombre de filtres passe-bande est élevé, plus l'analyse du signal sera fine.

À titre d'exemple, considérons un signal appelé bruit blanc, caractérisant une source d'information aléatoire. Son spectre de densité de puissance est donné en figure 6b.

Le spectre de densité de puissance du signal (aléatoire) bruit blanc montre clairement que toutes les fréquences ont un poids égal dans la composition du signal. Il n'y a donc aucune information. Cela est comparable, dans le cas numérique, avec la situation d'un signal binaire aléatoire, dans lequel les probabilités *P[bit vaut 1]* et *P[bit vaut 0]* sont égales à 0.5.

3. Les propriétés de la voix

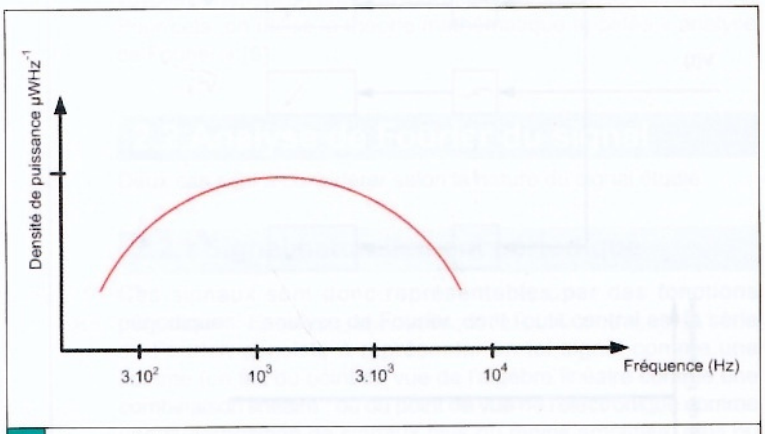
Nous allons maintenant nous intéresser à la voix proprement dite, en utilisant les concepts présentés précédemment.



6a & 6b *Analyseur de spectre et spectre de densité de puissance d'un signal bruit blanc*

3.1 Les Formants de la voix

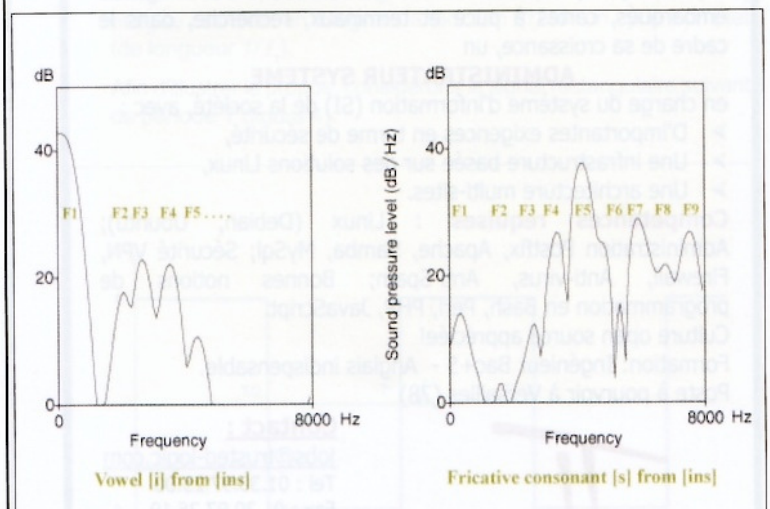
Tout d'abord, le spectre (de densité de puissance) de la voix, exprimé en $\mu Watt$ par Hertz (μWHz^{-1}) est donné en figure 7.



7 *Spectre de la voix*

On remarque que la voix se distingue par un spectre limité dans un intervalle de fréquences allant de 300 à 4 kHz (précisons que le spectre de la figure 7 est représenté en échelle logarithmique). Autrement dit, les fréquences supérieures à 4 kHz et celles inférieures à 300 Hz apportent seulement une contribution très faible, voire négligeable dans la composition du signal.

Pour étudier plus précisément la voix et identifier les composantes en fréquence les plus représentatives d'un signal donné, on établit un spectrogramme (voir un exemple avec la figure 8). Ce type de courbe caractérise certains pics de fréquences appelés « formants ». La composition en termes de formants est caractéristique d'un signal sonore donné. Cette composition spécifique permet ainsi de classer les différents types de signaux.



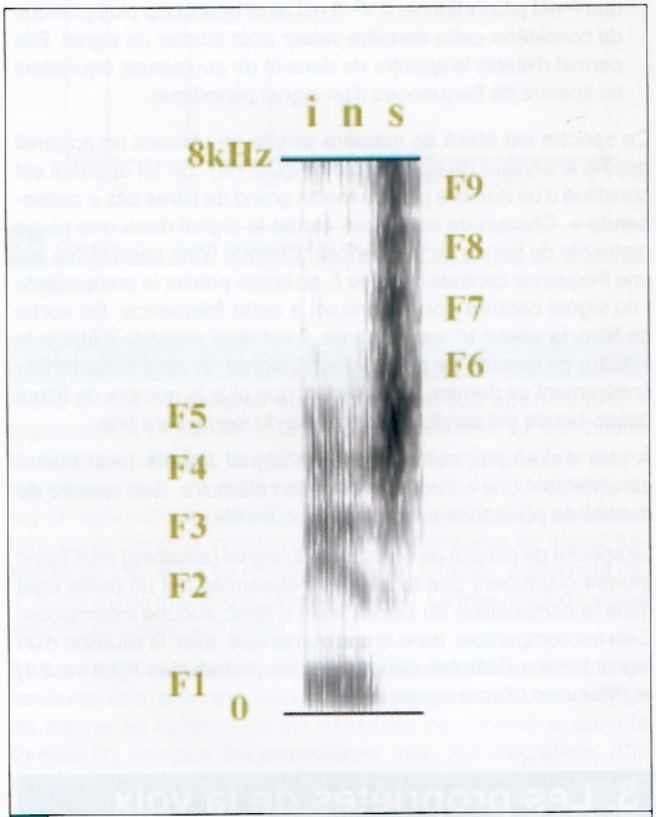
8 *Spectrogramme du signal sonore [ins] (à gauche le son associé à la voyelle [i] dans ce son ; à droite, celui de la consonne fricative [s]* ***[N4]*** *; en ordonnée est exprimée la variation de densité de puissance en dB/Hz).*

À titre d'illustration, considérons le signal correspondant au son [ins] en anglais, dont le spectrogramme est donné en figures 8 et 9 **[7]**. Le signal correspondant à la voyelle [i] comporte cinq formants étiquetés F1 à F5. Quatre autres formants existent entre 5000 Hz et 8000Hz dans le son du [i], mais leur contribution au signal est trop faible pour être visible sur le spectre (et donc être audible). Cette situation est inversée dans le cas de la lettre [s] où les fréquences F4-F9 sont, elles, fortement représentées, alors que les fréquences F1-F3 sont négligeables.

Les lettres sont ainsi classées en fonction de leurs principaux formants. Le tableau suivant donne quelques exemples pour quelques voyelles de la langue française.

Voyelles	Formant f_1	Formant f_2
A	1000 Hz	1400 Hz
E	500 Hz	2300 Hz
I	320 HZ	3200 Hz
O	500 Hz	1000 Hz
U	320 Hz	800 Hz
y	320 Hz	1650 Hz

TI *Formants des voyelles de la langue française*



9 *Répartition temporelle des formants du son [ins] (temps en abscisse).*

3.2 Le pitch (ou fréquence de pitch)

C'est la fréquence *f* de vibration de la corde vocale, dont la valeur moyenne varie d'un individu à l'autre. Chacun d'entre nous parle dans un intervalle de deux octaves, centré autour de cette fréquence moyenne soit [*f*/2, 2*f*]. Pour l'homme, la valeur de *f* est en moyenne de 130 Hz, tandis que celle des femmes est de 260 Hz.

Outre la constitution d'un signal sonore en termes de formants et de *pitch*, bien d'autres paramètres interviennent, qui définissent de manière spécifique la coloration d'un signal sonore donnée et représente une signature unique pour chaque individu. La sécurisation d'un signal sonore se devra de considérer tous ces paramètres afin de limiter fortement l'intelligibilité résiduelle d'un signal protégé, en particulier dans le cas des techniques purement analogiques [2].

4. La transmission de la voix

Toutes les caractéristiques présentées ont permis de caractériser finement un signal vocal, et en particulier de montrer que pour l'essentiel, il était localisé dans une bande de fréquences de 300 Hz à 4000 Hz. En pratique, les contraintes et/ou possibilités industrielles en termes de transmission et de communication vont exploiter ces caractéristiques de la manière la plus optimale possible, notamment pour maximiser le taux de transmission et le nombre de communications simultanées. C'est ce que l'on nomme le multiplexage par fréquence partagée. Cela exploite le fait que les canaux de transmissions (téléphonie, câble, fibre optique…) possèdent une bande passante largement supérieure à celle de la voix humaine (laquelle est, en pratique, limitée à la bande 300 Hz – 3400 Hz soit une bande passante de 3100 Hz). Par exemple, un câble pour transmission longues distances travaille dans une bande de 3 Mhz soit 3000 kHz. Il est donc possible de faire passer simultanément 1000 signaux de 3 KHz de bande passante chacun. Il suffit de décaler la fréquence des signaux pour les transmettre en parallèle.

Le premier signal ira de 1 à 3000 Hz, le second de 3000 à 6000 Hz et ainsi de suite. À l'autre bout, un démultiplexeur séparera les différents signaux avant de les transmettre aux destinataires.

D'autres limitations peuvent intervenir, en particulier dans le domaine des ondes radio. Ainsi, le signal sonore peut être restreint à une bande de 2,4 kHz de large, pour éviter les interférences, lorsque l'on travaille dans le domaine des hautes fréquences (HF) entre 3 MHz à 30 MHz, laquelle est très utilisée. En revanche, dans le domaine VHF (*Very High Frequency*, de 30 à 300 MHz) ou UHF (*Ultra High Frequency*, de 300 MHz à 3 GHz), le signal sonore peut être traité dans une bande plus large de 10 kHz.

Toutes ces données et limitations concernant la transmission d'un signal sonore en fonction du support doivent être bien connues et prises en compte lors de la sécurisation de ce signal sous peine d'altérer soit la qualité de ce signal, soit sa sécurité. Ces aspects-là seront présentés dans [2,3].

Conclusion

Cet article a présenté les techniques de représentation et de traitement de la voix, hors de toute préoccupation de sécurité. Mais, la connaissance de ces techniques et leur bonne compréhension sont essentielles si l'on veut ensuite pouvoir sécuriser une communication vocale. En effet, protéger une telle communication est beaucoup plus complexe que pour un « simple » message, fait de données numériques. Que ce soit sa confidentialité, son intégrité et certains aspects liés à l'authentification, un message vocal contient en plus des informations utiles, des informations critiques, liées à l'émetteur lui-même, qu'il est essentiel de protéger également. C'est ce que nous verrons dans les deux prochains articles.

Notes

[N1] Rappelons qu'une fonction est dite « linéaire » si et seulement si $f(x + y) = f(x) + f(y)$ et si $f(a.x) = a.\ f(x)$.

[N2] Un fonction *f* est dite « périodique de période *p* » si et seulement si pour tout *x*, nous avons $f(x + p) = f(x)$. Autrement, dit la fonction se répète à l'identique selon un motif de longueur *p*.

[N3] La variable d'intégration est ici ω, car on fait varier $n\omega_1$.

[N4] Une consonne fricative est une consonne dont le son est produit par un resserrement du chenal expiratoire (par exemple les consonnes s, f ou z). Le son finalement produit dépend de la position et de la tension musculaire des lèvres et de la langue.

Références

[1] SHANNON (C. E.), *A mathematical theory of communications*, *Bell System Technical Journal*, pp. 379 – 423 (juillet) et pp. 623 – 656 (octobre), 1948.

[2] FILIOL (E.), « La sécurité des communications vocales (2) : techniques analogiques ». *MISC – Le journal de la sécurité informatique*, 2008. À paraître.

[3] FILIOL (E.), « La sécurité des communications vocales (3) : techniques numériques », *MISC – Le journal de la sécurité informatique*, 2008. À paraître.

[4] ROSIE (A.M.), *Information and communication theory*, 2nd édition, Van Nostrand Reinhold Ed. 1973

[5] GODEMENT (R.), *Cours d'algèbre*, Hermann, 1997.

[6] GASQUET (C.) et WITOMSKI (P.), *Analyse de Fourier et applications*, Dunod, 2003

[7] WOOD (S.), « *What are formants* ? », 2005 : **http://person.sol.lu.se/SidneyWood/praate/index.html**