misc

MULTI-SYSTEM & INTERNET SECURITY COOKBOOK

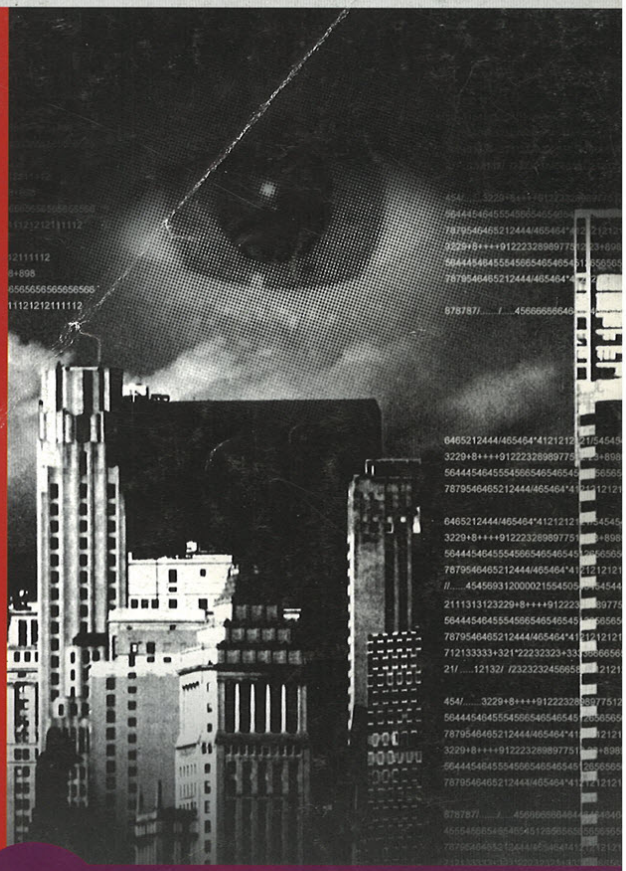
L 19018 - 22 - F: 7,45 € - RD
France Métro : 7,45 Eur - CH : 12,5 CHF
BEL. LUX. PORTCONT : 8,5 Eur - CAN : 13 $C
MAR : 75 DH

22 novembre décembre 2005

100 % SÉCURITÉ INFORMATIQUE

Superviser sa sécurité

- Organiser son système d'information
- Collecter les données
- Exploiter les informations
- Anticiper et réagir



PROGAMMATION

Dissimulation distribuée dans des fichiers Elf

FICHE TECHNIQUE

Reverse engineering facile avec DTrace

SCIENCE

Abuser les tests statistiques

N° 77
novembre
2005

Disponible en kiosque

GNU LINUX MAGAZINE / FRANCE

France Métro : 6,40€ - DOM 6,95€ - BEL : 7,30€ - LUX : 7,30€ - PORT. CONT. : 7,30€ - CH : 13FS - CAN : 12$ - MAR : 65DH
▶NOVEMBRE ▶2005 ▶NUMERO 77

08 ▶PEOPLE
Netfilter Workshop 2005, conférence utilisateurs et développeurs

22 ▶SYSADMIN
Dolibarr : et si vous utilisiez un ERP libre ?

60 ▶DÉVELOPPEMENT C/Tcl
Critcl, extensions Tcl en C à la volée

66 ▶DÉVELOPPEMENT PHP
Alertes météo avec PHP

70 ▶DÉVELOPPEMENT PERL
Construire des robots pour le web en Perl

Systèmes de Fichiers Chiffrés

Intégré depuis Linux 2.6.4, dm-crypt repose sur l'API cryptographique du noyau et les fonctionnalités du device mapper. Il offre une solution de chiffrement transparente, souple et puissante des périphériques bloc et des systèmes de fichiers

Développement Palm OS

Créez vos premières applications Palm OS depuis GNU/Linux avec

Administ

Nouvelle formule

Nouvelles rubriques

Nouvelle maquette

PEOPLE CONFÉRENCE

→ Netfilter Workshop 2005, conférence utilisateurs et développeurs

Éric Lebhard

EN DEUX MOTS Le quatrième atelier de travail de Netfilter, NFWS2005, a eu lieu à Séville du 3 octobre au 8 octobre 2005. Cet événement rassemble chaque année la plupart des développeurs de la couche pare-feu de Linux afin qu'ils puissent discuter des évolutions à venir. Pour la première fois lors de cette édition, une conférence utilisateurs (comprendre administrateurs système) a eu lieu lors des 2 premiers jours.

Astaro, Mara Systems, Balabit ou encore Ph, qui emploient ou finançent à elles seules une bonne partie des développeurs de Netfilter.

Ces fonds ont permis, d'une part, de distribuer des goodies très appréciés comme des T-shirts ou des pochettes et, d'autre part, de prendre en charge les frais de déplacement dans Séville et les repas de l'ensemble des participants de l'atelier des développeurs.

Unix pour le pare-feu, le feu étein au Kerosène et ailleurs (Photo réalisée sans trucage)

Rusty Russell, auteur de Iptables et auteur original de Netfilter, et Harald Welte, chef actuel de l'équipe de Netfilter (de gauche à droite)

Pablo Neira, développeur de l'outil conntrack et organisateur de NFWS2005, dans le hall de l'hôtel

(3-4 octobre) Conférence utilisateur

Une organisation impeccable

Le quatrième atelier de travail Netfilter s'est déroulé dans les locaux de l'Escuela Técnica Superior De Ingeniera Informatica de l'université de Séville. L'accueil sur place a été très chaleureux. L'organisateur Pablo Neira a en effet tenu une équipe de volontaires

La manifestation se déroulait sur plusieurs jours, et les participants venant d'un peu partout dans le monde, la question du logement était cruciale. Une partie d'un étage a donc été réservée pour loger les développeurs, ce qui a indéniablement apporté beaucoup au niveau du règlement des déplacements et de la convivialité.

Des exposés variés

Après l'ouverture de la conférence par des autorités politiques locales, nous avons assisté à une suite de conférences techniques.

La plupart d'entre elles étaient d'un niveau élevé ; les développeurs ou tout au moins des contributeurs étaient aux commandes, les explications étaient pour le moins complètes.

Du fait de la complexité des sujets abordés, je ne ferai ici qu'un compte-rendu partiel des exposés me concernant sur ceux m'étant apparus comme les plus significatifs.

NOTE

Le classement engage l'auteur qui s'est efforcé d'être le plus impartial possible.

Présentation de l'outil conntrack

Depuis très longtemps, une des plus fortes demandes des utilisateurs de Netfilter est l'ajout d'une interaction avec le système de suivi de connexions. Les requêtes les plus fréquentes concernent la modification depuis l'espace utilisateur d'entrées de la table de connexions ou encore une infrastructure cohérente de consultation du conntrack.

Cela est maintenant possible, grâce à la bibliothèque libconntrack et à l'outil conntrack présentés à la conférence par leur développeur Pablo Neira.

La commande conntrack permet d'ajouter, modifier ou encore supprimer des entrées de la table de connexions. Une deuxième option permet tout simplement d'afficher la

Bien que semblant triviale, cette fonctionnalité permet de régler un problème de conception : l'utilisation du pseudo-fichier "/proc/net/ip_conntrack" employé jusqu'ici à cet effet avait en effet tendance à bloquer la machine lors de sa consultation.

Enfin, un mode passif permet de recevoir en espace utilisateur les événements issus du système de suivi de connexions. Cet outil s'appuie sur des nouvelles fonctionnalités qui ont été incorporées au noyau 2.6.14.

La première ligne définit un ensemble appelé « spammers ». Toute machine ajoutée à cette ensemble y reste une semaine (le paramètre timeout est précisé en secondes). La deuxième permet de peupler l'ensemble « spammers ».

La machine 4DRECYPOT est en fait une adresse IP qui n'est PAS pour aucun domaine. Toute machine se connectant au port 25 du serveur est donc vraisemblablement animée d'intentions peu louables et voit donc son adresse IP ajoutée à l'ensemble « spammers ».

La troisième ligne enfin permet de réinitialiser (comprendre 10 min par mail) les machines de l'ensemble « spammers » lorsqu'elles tentent d'envoyer un mail vers les serveurs mail hébergés derrière le pare-feu.

Ipset est donc un outil extrêmement intéressant qui permet de plus d'accroître considérablement les performances de Netfilter au prix toutefois d'un accroissement de la complexité de génération des règles.

L'espace utilisateur pour les développeurs noyau

L'exposé de Rusty Russell, ou plutôt le show à l'américaine de l'ouverture, a été un des grands moments de cette conférence.

Il avait comme d'Australie des Caramels Koala afin, de les lancer à ceux qui, dans la foule, répondaient

Sommaire

ORGANISATION 4 → 10

> Une approche globale de la sécurité de l'information : théorie et pratique

DROIT 12 → 17

> La régulation des systèmes d'information en Chine

CHAMP LIBRE 18 → 21

> CVSS : un systhème en devenir ?

DOSSIER 22 → 63

Superviser sa sécurité

> Organiser la supervision de la sécurité informatique / 22 → 30

> Collecte d'informations / 31 → 36

> Des outils libres pour superviser la sécurité / 37 → 46

> La gestion des correctifs de sécurité / 48 → 55

> Reporting et procédures de réaction / 56 → 63

PROGRAMMATION 64 → 69

> DHIS Comme Distributed Hidden Storage

FICHE TECHNIQUE 70 → 75

> Le reverse engineering facile avec DTrace

SCIENCE 76 → 80

> La simulabilité des tests statistiques

> Abonnements et Commande des anciens Nos / 11/47/81

A CONÇU LE DOSSIER :

- Nicolas Fischbach

ONT RELU CE MAGAZINE :

- Axelle Apvrille
- Guillaume Arcas

Édito

L'eau, une ressource précieuse

Je passe toujours un temps non négligeable sur mes éditos : rassembler les nouvelles des uns et des autres pour tenir ma rubrique « carnet rose » à jour, passer des messages personnels, pousser des coups de gueule ou m'essayer tant bien que mal à la contrepèterie, au calembour ou à l'acrostiche. Cette fois, ce sera donc une allégorie écologique.

Comme l'an passé, me voici en Afrique pour les Rencontres Africaines des Logiciels Libres. Cette fois, nous sommes à Libreville au Gabon et l'ambiance est tout aussi studieuse et détendue. Pour ceux qui ne connaissent pas ce type d'ambiance, venez à SSTIC : certes, Rennes est plus humide que Libreville. Quoique, vu le temps qu'on a depuis notre arrivée... Et puis lors des éditions précédentes de SSTIC, nous avons toujours eu la chance d'avoir un temps exceptionnel, même si nous ne pouvions pas réellement en profiter, coincés au fond de l'amphi. Vraiment, je ne comprends pas pourquoi les gens ont une telle image de la Bretagne (remarquez la subtilité du rédac chef qui cherche à préserver ses ventes et sa tranquillité lors de son prochain séjour dans cette charmante province). À la lecture de cette pittoresque introduction, vous vous demandez probablement qu'est-ce qui me fait associer Libreville et Rennes, l'Afrique et la Bretagne, dans un même paragraphe. La réponse tient en quelques lettres : l'eau.

Certaines régions du monde ont une conscience aiguë de l'importance stratégique de cette ressource. Pour d'autres, c'est simplement une question de survie, ni plus ni moins. Enfin, dans certains endroits, il s'agit juste d'un consommable. Quoi qu'il en soit, c'est une évidence pour tous que l'eau est une denrée précieuse, voire critique. Devant la criticité évidente de l'eau, les associations, entreprises, organisations et autres pays se sont structurés pour tenter de la protéger. Cela se fait (presque) spontanément. En revanche, dans le domaine de la sécurité de l'information, nous en sommes encore aux balbutiements, à tenter de vaguement comprendre comment tout cela fonctionne. Qui est en mesure de donner une définition valable de « système d'information » ?

On voit des groupes en tout genre, qui pondent des rapports fleuve disant tout et son contraire, et autres experts autoproclamés, sans doute issus de la « bulle » (pas d'eau, internet), sans oublier que la moindre de leur prestation est forcément salée. Dans un genre gâchis différent, je suis effaré de voir les océans de « jeunes » qui commencent à sortir des multiples formations du domaine. Certains sont particulièrement impressionnants et se retrouvent malheureusement à faire de l'admin Windows en environnement netware, et dès qu'ils commencent à signaler quelques problèmes, on les noie sous des réunions vaseuses. De nombreux éléments sont maintenant réunis pour que nous disposions d'un vrai savoir-faire susceptible de nous permettre de rivaliser avec nos concurrents et néanmoins partenaires (entreprises ou états). Toutefois, tout évolue encore dans un certain marasme, au gré des courants (d'influence) et marées (voire marais). Ce qui manque à certains dirigeants est une vision claire du domaine. Ils ne voient que le petit bout de l'iceberg, juste quelques risques qu'ils encourent, mais pas les plus insidieux, ni les avantages qu'ils peuvent tirer. Ils restent dans une position attentiste, pendant qu'ailleurs les autres ne se gênent pas pour tracer les imprimantes [1], scanner complètement la mémoire des ordinateurs [2] ou collecter des informations [3] avec un cynisme glaçant. Est-ce parce que d'autres ont ces mauvaises pratiques (voire d'autres qui sont pires) que nous devons faire de même ? Certainement pas. Mais ne rien faire ne nous permettra pas de nous protéger. Enfin, vous l'attendez tous, le carnet rose : félicitations à Simon, et surtout bon courage à sa (nouvelle) femme. Courage également à Nico B. pour les quelques futures années d'angoisse à venir, suite à la naissance de sa fille Marine.

Assez d'histoires (d'eau bien sûr), les bonnes choses ont une fin et il est temps d'aller se réhydrater. Mais, assez d'eau pour cette fois :)

Fred Raynal

[1] *Secret Code in Color Printers Lets Government Track You Electronic Frontier Foundation* (EFF)
http://www.eff.org/news/archives/2005_10.php#004063
[2] *4.5 million copies of EULA-compliant spyware*
http://www.rootkit.com/blog.php?newsid=358
Evading hack detection mechanisms in online games
http://rootkit.com/newsread.php?newsid=360
[3] Politique de confidentialité des données chez Google
http://www.google.com/privacypolicy.html

Une approche globale de la sécurité de l'information : théorie et pratique

« Beaucoup de systèmes d'information n'ont pas été conçus sûrs. La sécurité qui peut être obtenue par des moyens techniques est limitée, et elle devrait être soutenue par des procédures et une organisation appropriées. »

Introduction de l'ISO/IEC 17799:2005(E)

Cet article présente une approche globale de la sécurité de l'information, d'où son titre ;-)... Après une courte introduction, nous verrons deux approches, les meilleures pratiques et les normes, pour appréhender les idées principales de la sécurité de l'information sans partir d'une page blanche, puis nous mettrons en lumière quelques éléments pragmatiques de son implémentation au sein d'une organisation.

1. Le processus de la sécurité de l'information

Tout est dit dans l'introduction de l'ISO 17799 citée ci-dessus : la sécurité de l'information (SI) n'est pas seulement une problématique technique. Certaines entreprises sont désormais conscientes que quelques briques technologiques, logicielles et matérielles, ne suffisent plus à protéger leurs informations critiques, d'ailleurs nous devrions dire « ne suffisent pas » car ils n'ont jamais suffi. Ces entreprises se tournent désormais vers le management de la sécurité, dans une approche globale, aussi bien organisationnelle, technologique que juridique.

Rappelons, que l'information n'a de valeur que lorsqu'elle est exploitée, traitée et souvent échangée avec d'autres entités internes ou externes. L'environnement dans lequel l'information évolue et est manipulée est donc complexe. Il est composé d'hommes et de relations humaines, de processus, d'éléments technologiques et d'exigences juridiques, tout ceci en mutation constante. Vouloir s'attaquer à la sécurisation de l'information implique l'obligation de prendre en compte tous ces paramètres. Il en est de même pour la sécurisation des systèmes qui participent à l'exploitation de cette information.

La SI se décline donc d'un point de vue humain, l'homme en tant qu'individu mais aussi au niveau des structures et des organisations qu'il met en place. Elle doit également être présente au niveau des processus métiers de l'entreprise et des procédures associées. Enfin, on la retrouve au niveau des éléments techniques constituant le système d'information.

De plus, l'environnement de l'information évolue sans cesse. Il sera donc nécessaire de vérifier chaque jour, de surveiller en continu l'adéquation des solutions trouvées hier à la problématique de la SI, et estimer ce qu'elles donneront demain. On voit donc que l'on entre dans un véritable processus de la SI, processus d'amélioration continue, consistant à identifier et mettre en place des mesures préventives et correctives, à surveiller leur performance et à corriger les écarts qui ne manqueront pas d'apparaître. On est loin du simple achat d'un pare-feu supplémentaire...

2. Ne pas réinventer la roue... de Deming

Devant l'ampleur et la complexité du processus de la sécurité de l'information, une aide pour sa mise en œuvre est souvent recherchée. Deux voies sont alors classiquement suivies, seules ou de façon complémentaire : les meilleures pratiques et les normes. Par la suite, nous allons classifier ces démarches possibles et les expliciter ensuite.

2.1. Classification des approches

Une approche inspirée de meilleures pratiques permet d'espérer bénéficier de l'expérience, des succès et aussi des erreurs de pairs, qui ont déjà dû traiter les mêmes sujets. Pour une entreprise, cette approche lui permet également de se comparer aux pratiques vraisemblablement mises en œuvre par les autres, en espérant faire ce qu'il faut pour être en sécurité, sans en faire trop, et donc sans investir dans des mesures moins rentables.

D'un autre côté, l'avantage des normes est d'être... des normes justement, qui constituent, enfin peut-on l'espérer, un consensus, voire au pire un compromis, entre experts internationaux sur la meilleure façon de traiter un sujet.

Les meilleures pratiques et les normes abordent la plupart du temps la problématique de la SI avec des orientations différentes. Schématiquement, on peut classifier les approches possibles comme suit :

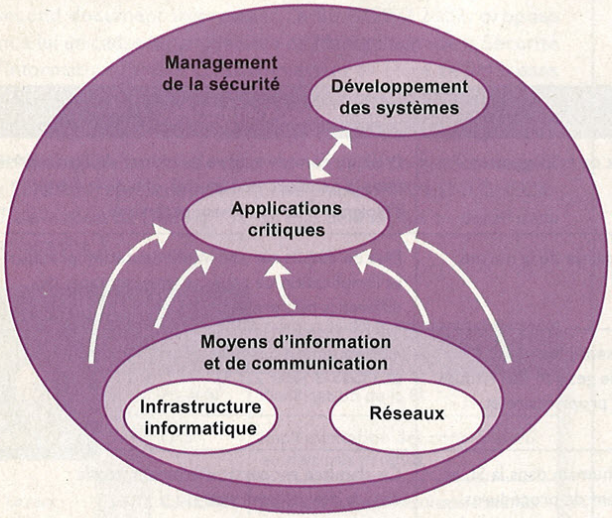
TABLEAU 1

Orientation de l'approche	Meilleures pratiques	Normes
Gestion des risques	EBIOS[1] MEHARI[2] OCTAVE[3]	
Processus	ITIL[6]	ISO 9001[4] ISO 13335-2, 13335-3[4] BS 7799-2[5]
Contrôles / Mesures de protection	COBIT[7] IT Baseline Protection Manual du BSI [8]	ISO 17799[4] ISO 13335-4
Produits		ISO 15408[4]

S'appuyer sur de telles approches, quelles qu'elles soient, permet de s'assurer d'une certaine exhaustivité, cohérence et homogénéité dans sa démarche, et fournit également des éléments communs (vocabulaire, concepts, ...) à tous les intervenants dans le projet : décideurs, utilisateurs, personnels de l'informatique, sous-traitants, fournisseurs, partenaires, etc.

Olivier Busolini
olb@netexpert.ch

TABLEAU 2



Décomposition d'un aspect en domaines et sections

Aspect « Management de la sécurité »		
Area		direction haut niveau
	Section	engagement des dirigeants
	Section	politique de sécurité
	Section	acceptation des utilisateurs
Area		organisation de la sécurité
Area		besoins de sécurité
Area		environnement sécurisé
Area		attaques malicieuses
Area		sujets spéciaux
Area		revue du management

Les cinq aspects de la SI selon l'ISF.

2.2 Meilleures pratiques

De nombreux guides de meilleures pratiques existent. Ils peuvent être nationaux, comme certains composants de la série 800 du NIST[11], ou réalisés par l'industrie, comme les publications de l'ISF (*Information Security Forum*)[10], de l'ISACA (*Information Systems Audit and Control Association*)[13].

Prenons comme exemple le « Standard de bonnes pratiques pour la sécurité de l'information » de l'ISF. Ce document est destiné aux entreprises de moyenne à grande taille, de n'importe quel secteur. Selon l'ISF, une entreprise devrait avoir comme objectif final la mise en œuvre de toutes les mesures mentionnées dans son standard afin de limiter les risques pesant sur l'information à un niveau acceptable.

L'ISF a choisi de découper la SI en cinq aspects comme montré dans la figure suivante : Les applications critiques se trouvent au centre de cette approche et reposent sur les soubassements techniques de l'infrastructure informatique et des réseaux. Le développement des systèmes aborde la façon dont de nouvelles applications sont créées, et le management de la sécurité explicite les mesures de gouvernance et de contrôle.

Chacun de ces aspects est décliné en domaines (area) qui sont à leur tour déclinés en sections. Enfin, chaque section possède un principe qui spécifie ce qui doit être fait ainsi qu'un objectif qui indique les raisons pour lesquelles quelque chose doit être fait. Ces principes et objectifs haut niveau dans le domaine de la SI sont suivis de mesures à mettre en œuvre pour atteindre les objectifs.

Même si l'approche des meilleures pratiques consiste globalement à puiser dans un catalogue de mesures celles qui sont le plus adaptées au contexte de l'organisation, elle n'économise pas l'étude des éléments particuliers à chaque organisation, afin de choisir les mesures les plus pertinentes et de les adapter aux besoins spécifiques de chaque situation. Il ne s'agit pas de remplacer l'analyse par la copie.

2.3 Normes et standards

L'approche normative privilégie l'utilisation de normes ou standards reconnus afin de mettre en œuvre une SI globale dans son organisme. Parmi l'ensemble des normes existantes, deux s'intéressent tout particulièrement à ce processus de la SI : la paire BS 7799-2 / ISO 17799 et l'ISO 13335.

Pour schématiser, on peut dire que les normes BS 7799-2 et ISO 17799 proposent une approche à la fois technologique, axée sur ce qui doit être fait (ISO 17799) et organisationnelle (BS 7799-2).

A contrario, l'ISO 13335 se concentre sur la manière de faire, d'un point de vue organisationnel, juridique et technique. Elle part des concepts et des processus de management de la sécurité pour aboutir in fine à des choix de mesures de protection techniques.

Nous allons les présenter rapidement, sans pour autant tomber dans un plagiat de leur contenu.

2.3.1 ISO 17799

ISO a adopté le BS 7799 (et pas BS 7799-2 !) sous le nom de ISO 17799:2000 « Code de pratiques pour la gestion de la sécurité de l'information » en procédure « *fast track* » (c'est-à-dire suite à peu de négociations entre les différents pays membres de l'ISO ...) en 2000. Il propose des recommandations pour assurer la sécurité de l'information, sous la forme d'objectifs de contrôles ou de mesures de sécurité répartis en 10 domaines décrits plus bas.

Dans le cadre d'un processus d'amélioration de ses documents, l'ISO 17799 a été mis à jour en juin 2005. Cette nouvelle version ISO 17799:2005(E) propose des moyens de contrôle, des mesures dans le but de réduire les risques qui pèsent sur la SI. Ces contrôles sont répartis par domaines, un objectif de sécurité est énoncé pour chaque domaine, définissant le résultat qui doit être obtenu, et des contrôles sont proposés afin d'atteindre l'objectif de contrôle recherché.

TABLEAU 3

	Domaine	Accent du domaine	Modifications apportées
1	Politique de sécurité	Être plus encore un document de management stratégique de l'entreprise	Prise en compte accrue du métier de l'entreprise dans la politique de sécurité, afin de la faire s'éloigner d'un document technique
2	Organisation de la sécurité de l'information	L'emphase est mise sur cette partie de la norme	Plus de détails sur son implémentation pratique et l'implication de tous les représentants des différents métiers dans la SI
3	Classification et contrôle des ressources	Identification des ressources sensibles comme étape majeure du processus de gestion des risques et importance des notions de propriétaire et utilisateur des ressources	Augmentation des liens vers l'ISO 13335-1 et 13335-3 (voir 2.3.3)
4	Ressources humaines	Importance de la gestion de l'humain dans la SI, et en particulier, le développement de procédures de sécurité lors des mouvements des personnels (arrivée, mais surtout mutations et départs) Rôle des dirigeants de l'entreprise, dont la sécurité de l'information est une des responsabilités à part entière, qu'ils doivent assumer et promouvoir	Ce chapitre est particulièrement étoffé dans la nouvelle version
5	Sécurité physique	Traitement de la sécurité physique de l'information ainsi que de la sécurisation des locaux et des équipements	Modifications mineures
6	Management des communications et des opérations		Plus de détails concernant le sujet des tierces parties (sous-traitants, consultants, etc.) et le monitoring de la sécurité (qui quitte le chapitre « contrôle d'accès » pour celui-ci) dans les phases de vérification et d'amélioration du système de management de la sécurité de l'information
7	Contrôle d'accès	Traitement du contrôle d'accès au niveau applicatif, réseau et système ainsi que des problématiques liées à la mobilité et au télétravail	Modifications mineures
8	Acquisition, développement et maintenance des systèmes d'information		Ajout d'un paragraphe spécifique sur la gestion des vulnérabilités techniques
9	Gestion des incidents de sécurité de l'information	Ce chapitre a été créé dans la nouvelle version de l'ISO sur la base du paragraphe 8.1.3 de l'édition précédente	Développement plus en profondeur
10	Plan de continuité	Définition du contenu d'un plan de continuité, incluant la sécurité de l'information	Modifications mineures
11	Conformité	Prise en compte des lois, des règles et standards de l'organisation, et des audits permettant de vérifier la conformité des mesures mises en place	Modifications mineures

En résumé, l'ISO 17799:2005 prend un peu de recul avec les technologies et les techniques, apporte une mise à jour des contrôles présents dans la version précédente, ajoute des contrôles relatifs au management de la sécurité, et accentue l'importance de la gestion des risques. Le tableau 3 met en lumière le fond des modifications apportées par la nouvelle version aux différents domaines de la norme.

2.3.2 BS7799-2

Le second document intéressant, le BS 7799-2:2002, propose quant à lui un cadre ou un Système de Management de la Sécurité de l'Information (SMSI). C'est en mettant en place ce SMSI dans son organisation que l'entreprise va pouvoir gérer sa SI selon une méthode qui se veut exhaustive et reconnue.

Le SMSI s'appuie principalement sur un cycle de gestion de la SI dit « PDCA », représentant les quatre phases PLAN-DO,CHECK-ACT de la roue de Deming [9], tel que définis plus en détail dans le paragraphe 3.

TABLEAU 4

La roue originale de Deming

Étape		Activité
1	**PLAN**	Planification de la SI
2	**DO**	Implémentation des contrôles et/ou mesures retenus
3	**CHECK**	Vérification des résultats obtenus
4	**ACT**	Amélioration du processus global, et retour à l'étape 1

2.3.3 ISO 13335

L'ISO 13335 intitulé « Management de la sécurité des technologies de l'information et de la communication » offre une autre approche du management de la sécurité des technologies de l'information. Il aborde le sujet à différents niveaux dans ses cinq rapports techniques 13335-1 à 13335-5.

Il propose d'abord une approche stratégique (que faire ?) dans ses parties 13335-1, qui présente les concepts et modèles de la sécurité des technologies de l'information, et 13335-2 qui propose une approche du management et de la planification de la sécurité des technologies de l'information. Ces deux parties sont destinées à donner aux dirigeants et aux responsables de la SI d'une organisation, les concepts et méthodes relatifs au management de la sécurité des technologies de l'information.

On trouve ensuite une approche pratique (comment faire ?) de la mise en œuvre du management de la sécurité des technologies de l'information. Dans la partie 13335-3 « Techniques pour le management de la sécurité des technologies de l'information », on trouve des moyens de réaliser les actions énoncées dans la partie 2.

La partie 13335-4 « Sélection de mesures de protection » propose un processus de sélection de mesures de protection adaptées aux besoins de sécurité qui auraient été mis en lumière lors d'une analyse des risques menée en partie 3. Même si ces mesures de protection sont générales, des tableaux renvoient à des catalogues tiers de mesures de protection détaillées et de bonnes pratiques, comme :

- ISO 17799 ;
- ETSI *Baseline Security Standard* ;
- IT *Baseline Protection Manual du Bundesamt für Sicherheit in der Informationstechnik* ;
- NIST *Computer Security Handbook* ;
- *Medical Informatics : Security Categorisation and Protection for Healthcare Information Systems* ;
- TC68 *Banking and Related Financial Services* ;
- *Protection of Sensitive Information not covered by the Official Secret Act* ;
- *Canadian Handbook on Information Technology Security.*

Enfin, la partie 13335-5 offre un guide pour le management de la sécurité des réseaux, mais sans entrer trop dans les mesures techniques.

Révision

À l'image de ce que nous avons vu pour l'ISO 17799, l'ISO 13335 est en cours de révision. Les rapports techniques 2, 3, 4 et 5 sont assez anciens et le tout va être refondu en deux parties :

- une norme ISO 13335 sur la gestion de la sécurité des technologies de l'information et des communications, en deux parties : la 13335-1 « Concepts et modèles pour la gestion de la sécurité des technologies de l'information et des communications » publiée fin 2004 et la 13335-2 « Gestion de risque de la sécurité de l'information » ;
- des rapports techniques, les 13335-4 « Sélection de sauvegardes » et 13335-5 « Guide pour la gestion de sécurité du réseau ».

(Voir tableau 5, page suivante.)

Nous avons donc vu qu'il existe plusieurs méthodes permettant de guider l'entreprise dans la mise en œuvre d'un processus de sécurité de l'information. En particulier, les BS 7799-2 et ISO 17799 peuvent servir de guide pour réaliser un système de management de la sécurité de l'information. Intéressons-nous maintenant à son implémentation pratique.

3. Un système de management de la sécurité de l'information

À l'image de beaucoup d'entreprises, choisissons le standard BS 7799-2 comme guide pour réaliser un système de management de la SI. Dans les paragraphes qui suivent, nous allons d'abord définir la structure d'un tel système avant d'aborder son implémentation.

3.1. Structure du système de management de la SI

Comme cela a été dit plus haut, il s'agit de mettre en place au sein de l'entreprise un cycle de gestion de la SI en quatre étapes « PLAN – DO – CHECK – ACT ».

À la première étape, il s'agit d'établir les fondements de la SI de l'entreprise, et en particulier la politique de sécurité ainsi que les processus et procédures de gestion de la sécurité de l'information afin d'atteindre les objectifs de sécurité identifiés par l'organisation. Cette étape va également contenir la préparation des éléments nécessaires au pilotage du SMSI, c'est-à-dire la sélection des indicateurs adéquats pour suivre le niveau de sécurité dans les différents domaines traités.

Dans la deuxième étape, il va falloir mettre en œuvre les mesures de sécurité choisies dans l'étape précédente. Il est possible de s'appuyer sur des mesures issues de catalogues de bonnes pratiques ou de l'ISO 17799. Ces mesures seront de tous ordres : organisationnelles, humaines, procédurales, techniques, etc. Leur propriété fondamentale commune est de participer à la réduction des risques identifiés précédemment. En ce qui concerne les indicateurs, on veillera à définir leur métrique, quoi mesurer et comment, et à les alimenter.

Il convient ensuite de « refermer la boucle », donc de mesurer les résultats atteints et d'apporter les corrections nécessaires. C'est le but des étapes « CHECK » et « ACT ».

L'appréciation des performances des moyens mis en œuvre se fonde sur l'exploitation des indicateurs et leur évolution. Ces indicateurs pourront également servir à montrer l'évolution du système de management de la SI, aux dirigeants par exemple. Enfin, la dernière étape consiste à apporter les corrections nécessaires aux mesures mises en place afin d'amener les performances au niveau objectif qui a été déterminé par l'organisation.

Le système de management de la SI étant un processus bouclé, il est important de réexaminer le contenu de chaque étape régulièrement, à un rythme annuel voire tous les deux ans. Pour cela, il est primordial d'avoir établi et maintenu à jour une documentation exhaustive de chaque étape. Cette base documentaire enregistre les directions choisies, les options qui ont été laissées de côté, les décisions et toutes les justifications nécessaires. Même si ce formalisme est lourd lors de sa première mise en place, il prend toute sa valeur lors des bouclages successifs, car il permet de ne plus agir que par différence, et non pas de refaire tout le travail à chaque itération.

3.1. Implémentation du système de management de la SI

Nous venons de définir la structure d'un système de management de la SI tel que le propose le standard BS 7799-2. Intéressons-nous maintenant à la façon de mettre en œuvre un tel système de management dans une entreprise.

La première phase est la définition exacte des informations qui vont être prises en compte. Choisir un périmètre trop important risque de conduire à un projet sans fin, alors qu'un périmètre trop restreint réduit d'autant l'intérêt de la démarche. Ensuite, il faudra identifier les objectifs de sécurité dans le périmètre retenu, c'est-à-dire savoir comment on veut protéger quelle information, contre qui ou quoi. Il s'agit donc de réaliser une analyse de risques pesant sur le patrimoine informationnel de l'entreprise. Cette analyse de risques est requise par le BS 7799-2, même si le document ne dit pas comment elle doit être réalisée. À chaque entreprise donc de choisir sa méthode, parmi les très nombreuses existantes.

Sans vouloir entrer dans une étude exhaustive des différentes méthodes, disons globalement que l'analyse de risques selon l'ISO 17799 et le BS 7799-2 se déroule selon les étapes suivantes :

1 D'abord, il s'agit d'identifier les informations de l'entreprise les plus sensibles et les processus de traitement de l'information afférents. Le terme de processus doit être pris au sens le plus large, par exemple le processus de gestion des ressources humaines ou celui du recrutement de personnel.

2 Ces processus sont mis en œuvre par des ressources humaines, organisationnelles et techniques qui doivent être identifiées à leur tour.

3 Nous allons ensuite analyser et apprécier le risque qui pèse sur ces ressources, c'est-à-dire identifier qui ou quoi pourrait mettre en œuvre quelle attaque, pour exploiter quelle vulnérabilité des ressources, avec quelle probabilité et quel impact sur l'entreprise.

4 Les risques identifiés sont alors traités : nous pouvons les accepter et ne rien faire, ou les transférer en externalisant une activité par exemple, voire les réduire en agissant sur leurs

TABLEAU 5 — Des normes intéressantes en sécurité de l'information

Appellation actuelle	Appellation future	Titre
ISO/IEC 17799:2005	ISO/IEC 27002 (prévu 2007)	• Technologies de l'Information • Techniques de sécurité • Code de pratiques pour la gestion de la sécurité de l'information
BS 7799-2:2002	ISO/IEC 27001 (prévu fin 2005)	• Systèmes de management de la sécurité de l'information • Spécifications et guide d'utilisation
BS 7799-2:2002 Annexe B	ISO/IEC 27003 (prévu 2006-2007)	• Standard de métriques et de mesures d'un SMSI
ISO/IEC 13335	ISO/IEC 13335 refondu	• Technologies de l'Information • Techniques de sécurité • Management de la sécurité des technologies de l'information et de la communication

TABLEAU 6

Stade	Caractéristiques	Description
1	incompétents en sécurité et inconscients de l'être → *Ils ne savent pas qu'ils ne savent pas.*	C'est le cas de l'immense majorité des personnes, même si la médiatisation des virus et autres « piratages » leur fait connaître petit à petit le monde de la SI.
2	incompétents en sécurité mais conscients de l'être → *Ils savent qu'ils ne savent pas.*	Cela demande à certains d'accepter, de faire le deuil de compétences qu'ils croyaient avoir. En tout cas, c'est le premier objectif d'une campagne de sensibilisation.
3	compétents en sécurité et conscients de l'être → *Ils savent qu'ils savent.*	C'est l'objectif principal d'une campagne de sensibilisation, et un stade tout à fait acceptable pour une organisation. Donner une bonne compétence en sécurité à tous les acteurs nécessite une large diffusion des pratiques requises par l'entreprise. Cela signifie également que ces bonnes pratiques ont été définies et formalisées au préalable.
4	compétents en sécurité et inconscients → *Ils ne savent plus qu'ils savent.*	Il s'agit de l'état idéal, où la sécurité fait tellement partie de la culture de chacun qu'elle est pratiquée inconsciemment.

origines ou leurs conséquences, c'est-à-dire en se fixant des objectifs de sécurité.

5 Enfin, nous allons choisir des mesures à mettre en œuvre pour atteindre ces objectifs de sécurité. Elles vont être le fondement de la politique de sécurité de l'entreprise.

6 Il s'agit ensuite de faire le bilan des mesures de sécurité déjà en place dans l'entreprise. Un plan de travail sera alors mis en place pour réaliser les mesures manquantes.

5. Retours d'expériences

Parmi les enseignements d'implémentation de SMSI réussies, on peut identifier quelques éléments clé de succès, des facteurs déterminants sans lesquels un projet de SMSI n'a que peu de chances d'aboutir.

5.1 Engagement des dirigeants

Un premier facteur déterminant pour la réussite d'un projet de SMSI est l'engagement réel et affiché des dirigeants et des managers intermédiaires de l'organisation. Un projet de SMSI est une démarche de gestion des risques qui pèsent sur le patrimoine informationnel de l'organisation. À ce titre, il s'agit donc d'un projet stratégique pour l'organisation qui dépasse la seule compétence de la direction des systèmes d'information, et qui doit émaner, ou au moins recevoir le soutien et la participation, de l'exécutif.

De plus, les conséquences de la démarche dépasseront largement le simple cadre des systèmes d'information, et auront vraisemblablement un impact sur toute l'organisation. La démarche doit donc être soutenue au plus haut niveau afin d'obtenir les moyens humains et financiers nécessaires à sa mise en pratique.

5.2 Sensibilisation des utilisateurs

Pour obtenir le soutien et la participation du management, mais aussi de tous les acteurs de l'organisation, encore faut-il les avoir sensibilisés aux risques et aux enjeux de la SI. Il s'agit là d'une modification de leur sensibilité, de leur culture et de leurs habitudes, ce qui ne se fait pas en quelques jours. De véritables campagnes de communication doivent être organisées sur le thème de la sensibilisation à la SI, avec le soutien de professionnels de la communication.

Nous voyons une nouvelle fois que les personnes sont au centre de notre démarche. Après avoir converti les dirigeants ;-), attaquons-nous aux autres acteurs de l'organisation, à savoir les utilisateurs mais aussi une autre catégorie de personnels parfois difficiles, les responsables techniques des systèmes d'information. Les utilisateurs doivent être convaincus de l'utilité de faire des efforts au jour le jour, d'accepter les quelques lourdeurs de certaines mesures de sécurité, et de changer eux aussi leur culture et leurs habitudes. Souvent, ils n'acceptent pas facilement ces efforts. C'est dans ces circonstances que le soutien des dirigeants est nécessaire afin de motiver les utilisateurs, sans lesquels le SMSI ne pourra fonctionner.

Concernant les responsables techniques et autres administrateurs système ou réseau, nous avons souvent rencontré des problèmes liés à une mauvaise évaluation de leur part des problématiques de sécurité. Même si toute généralisation est fausse, risquons-nous à dire que les concepts de sécurité ne font pas toujours partie de leurs connaissances générales et cela souvent à l'opposé de leur croyance. Il faut alors montrer, et souvent prouver, à ces professionnels que leurs pratiques ne sont pas sécurisées comme elles devraient l'être, avant de leur proposer de nouvelles façons de mieux faire leur métier.

5.3 Classification selon les stades d'apprentissage

Pour résumer le travail de sensibilisation, d'information et de formation à faire, on peut s'inspirer des stades d'apprentissage de l'adulte et les calquer à notre problème. Nous allons donc devoir faire cheminer les acteurs dans le domaine de la SI du stade 1 au stade 3, voire au stade 4 : cf. tableau ci-dessus.

Attention tout de même, car parfois certaines personnes peuvent être au stade 1 et croire (ou vouloir faire croire) être au stade 4, comme quand elles expliquent que « les procédures relatives à la sécurité ne sont pas écrites car elles font partie intégrante de la façon de travailler de tout le monde … » ;-)

5.4 Prise en compte des objectifs métiers

Un autre facteur déterminant est la prise en compte des objectifs métiers de l'organisation dans la démarche. Il est souvent difficile mais fondamental de montrer en quoi les mesures qui font parties du SMSI ne sont pas des freins supplémentaires aux activités de l'entreprise, mais bien des éléments qui permettront de travailler mieux : plus vite, de façon moins risquée, voire d'économiser à court, moyen ou long terme des sommes importantes.

Nous imaginons bien le petit sourire que ces phrases pourront générer dans les rangs des plus « techniciens » d'entre nous. Plus haut, nous parlions d'un effort à faire pour inculquer un peu de culture sécurité dans les préoccupations des dirigeants et des utilisateurs. Nous parlons ici d'un autre effort que nous, les « techniciens », devrions faire pour comprendre un peu plus le fonctionnement des « autres » afin de présenter nos convictions sous la forme la plus adaptée possible, dans le but qu'elles soient comprises et acceptées de tous. C'est seulement au prix de cet effort que les mesures que nous pensons indispensables à la SI seront finalement implémentées.

5.5 Approche pragmatique

Enfin, terminons par quelques mots relatifs à un autre standard en préparation, l'ISO 27799. Ce document propose une approche pragmatique pour implémenter le BS 7799-2 dans le domaine hospitalier, en tenant compte d'erreurs et d'expériences passées. Il peut néanmoins être intéressant pour toutes les entreprises même hors du domaine de la santé, car il propose également une partie en forme de guide d'implémentation de la démarche BS 7799-2. Cette future norme préconise en particulier :

- la sélection d'un premier périmètre limité à une dizaine de processus, ayant prouvé qu'il permettait de réaliser concrètement un SMSI dans des délais raisonnables, puis une itération sur des périmètres plus vastes ou d'autres processus ;
- la mise en place d'une organisation humaine particulière afin d'implémenter un SMSI qui vient en complément de l'organisation humaine proposée par l'ISO 13335-2 pour gérer un SMSI ;
- l'utilisation d'un outil d'aide à la réalisation d'un SMSI et d'aide à l'analyse de risques.

6. Conclusion

La sécurité de l'information ne peut pas être traitée par une approche uniquement technique, comme c'est encore trop souvent le cas. Le niveau de sécurité de l'information adaptée à chaque organisation ne peut être atteint et conservé qu'en impliquant tous les acteurs et les processus de cette organisation.

À l'image de la qualité qui a transformé radicalement le fonctionnement des organisations dans les années 80, le patrimoine informationnel de toute organisation ne pourra être sécurisé qu'après un changement de culture et de méthodes de travail de cette l'organisation.

Un tel processus sera sans doute long à mettre en œuvre, mais des guides et des compétences existent pour aider chaque organisation dans sa mutation. Mais comme en montagne, le guide ne fait que donner la direction, c'est aux randonneurs de faire chaque pas.

Notes et liens

[1] EBIOS : Expression des Besoins et Identification des Objectifs de Sécurité, **www.ssi.gouv.fr/fr/confiance/ebios.html**

[2] Méthode MEHARI : **https://www.clusif.asso.fr/fr/production/mehari/**

[3] OCTAVE : *Operationally Critical Threat, Asset, and Vulnerability Evaluation*, **www.cert.org/octave/**

[4] ISO : *International Organization for Standardization*, **www.iso.org/**

[5] BSI : *British Standards Institution*, **www.bsi-global.com/Global/bs7799.xalter**

[6] ITIL : IT *Infrastructure Library*, **www.itil.co.uk/**

[7] COBIT : *Control Objectives for Information and related Technology*, **www.isaca.org/cobit.htm**

[8] BSI : *Bundesamt für Sicherheit in der Informationstechnik*, **www.bsi.de**

[9] La roue de Deming originale est formée des 4 étapes Plan-Do-Study-Act : **http://fr.wikipedia.org/wiki/Roue_de_Deming**

[10] ISF : *Information Security Forum*, **www.isfsecuritystandard.com/index_ns.htm**

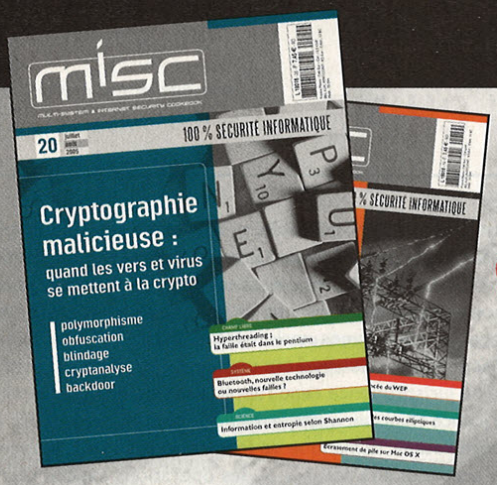
[11] NIST : *National Institute of Standards and Technology*, (**http://csrc.nist.gov/publications/nistpubs/**) qui a publié, entre autres, les documents suivants :

- 800-14 *Generally Accepted Principles and Practices for 14*
- 800- 12 *The computer Security Handbook*
- 800 800-18 *Guide for Developing Security Plans*
- 800 800-26 *Self Assessment Guide for IT Systems*

[12] ISACA : *Information Systems Audit and Control Association*, **http://www.isaca.org/**

Abonnez-vous à misc

MULTI-SYSTEM & INTERNET SECURITY COOKBOOK



soit 6 1 an de sécurité informatique numéros de Misc

= 33 €

44,70 €

France Metro

Offres de couplage possibles ! voir page xx

4 façons de vous abonner :

- par courrier postal en nous renvoyant le bon ci-dessous
- par le Web, sur **www.ed-diamond.com**
- par téléphone, entre 9h-12h et 14h-17h au 03 88 58 02 08
- par fax au 03 88 58 02 09 (CB)

Bon de commande à remplir et à retourner à :
Diamond Editions - Service des Abonnements/Commandes 6, rue de la Scheer B.P. 121 - 67603 Sélestat Cedex

Oui je souhaite m'abonner à Misc, 6 numéros

1 Voici mes coordonnées postales

Nom :

Prénom :

Adresse :

Code Postal :

Ville :

2 Je joins mon réglement :

Je règle par chèque bancaire ou postal à l'ordre de Diamond Editions*

Paiement par carte bancaire :

N° Carte :

Expire le : __/__/

Date et signature obligatoire : __/__/ 200_

3 BONNES RAISONS de vous abonner :

- Ne manquez plus aucun numéro !
- Recevez Misc tous les 2 mois, chez vous, ou dans votre entreprise.
- Economisez 11,7 €/an.

☐ Pour avoir un suivi par e-mail de vos abonnements, merci de nous indiquer votre adresse e-mail** :

**En application des articles 27 et 34 de la loi dite «Informatique et libertés» n° 78-17 du 6 janvier 1978, vous disposez d'un droit d'accès et de rectification aux données vous concernant.

*Pour les tarifs étrangers, consultez notre site : www.ed-diamond.com

LA RÉGULATION DES SYSTÈMES D'INFORMATION EN CHINE

L'Asie, 60% de la population mondiale, comptera plus de 240 millions d'internautes à la fin 2005 [1]. La Chine représente près de la moitié de ce potentiel, devenant ainsi le second marché mondial de PC, un parc de 100 millions de machines devant être atteint d'ici 2010. Le pays compterait aujourd'hui plus de 1000 FAI et hébergeurs, 10 000 fournisseurs de contenus et 100 millions d'utilisateurs [2]. Mais dans ce contexte de croissance, l'Etat chinois régule la mise en œuvre et l'utilisation des systèmes d'information, en s'appuyant sur deux instruments : la technologie et le droit.

I — Évolution de l'informatique et du droit

C'est à partir de 1978, date à laquelle Deng Xiaoping prit la tête du parti communiste chinois, que la Chine a commencé à se doter d'une véritable industrie de l'informatique, important matériels et logiciels américains et japonais pour se mettre à niveau. Depuis, la contribution des entreprises étrangères n'a cessé de croître au service du développement des infrastructures réseaux notamment : Nortel Networks, Sun Microsystems, 3COM... Le cœur d'Internet [3] repose d'ailleurs sur des technologies Cisco et l'implication de cette compagnie se poursuit dans le cadre du développement de l'Internet nouvelle génération [4]. En 1993, la Chine a initié le vaste programme « Projets Dorés », toujours en cours, visant à doter le pays des infrastructures de télécommunications et d'information chinoises nécessaires au développement économique : développement des réseaux d'information économique, réseau d'information des douanes, mise en relation des centres financiers, réseaux reliant 12000 PME et grandes entreprises, etc. Les infrastructures du pays se sont modernisées rapidement. La Chine a bouclé fin 2004 la connexion de 25 universités entre elles par un *backbone* nouvelle génération, Cernet2 (*China Education and Research Network* 2), premier réseau au protocole IPv6 avec des débits records de 40 Gb/s. La Chine offre ainsi l'image d'un pays à la pointe de l'innovation, avec sa Silicon Valley (Zhongguancun) ou encore son « corridor » stratégique [5] dans le domaine des télécommunications (Pékin – Shanghai – Guangzhou).

Parallèlement au développement économique, industriel et technologique, un renouveau dans le domaine juridique s'imposait. Au sortir de dix années de Révolution Culturelle, la Chine était dotée d'un appareil juridique hérité du système soviétique [6], incapable de répondre aux attentes du négoce international, des coopérations industrielles, de l'économie nouvelle désireuse d'attirer des capitaux étrangers. Le chantier était énorme. Furent publiés en 1979 le **Code Pénal**, en 1987 les **Principes Généraux du Droit Civil**. Mais c'est surtout à partir des années 90 que les lois majeures sont adoptées et qu'est amorcé un processus de codification. Au cours des 25 dernières années, la Chine a parcouru un chemin considérable, créant une législation de plus en plus complète. Les principes, textes de droit et leur application tels que conçus en Chine s'éloignent toutefois sur bien des points de nos conceptions européennes de la justice. Le morcellement du droit, avec ses normes nationales et locales qui parfois se contredisent, donne au système une certaine inconsistance et l'absence de jurisprudence rend l'application du droit incertaine. De cette justice chinoise, outre le maintien de la peine de mort, on peut encore retenir le manque de formation des juges, l'absence de droits accordés aux accusés ou les opérations spectaculaires sous formes de campagnes de « nettoyage » [7].

A partir du milieu des années 1990, la Chine n'a pas échappé à la frénésie de réglementation relative aux NTIC qui caractérise les pays occidentaux.

II — Régulation : contenus, acteurs, outils

Le pouvoir veut avoir la maîtrise des systèmes d'information. Pour cela, il dispose de deux moyens, l'un technologique, l'autre juridique (lois, régulation, autodiscipline, enregistrement, déclarations obligatoires, emprisonnement, l'ensemble créant un climat d'autocensure). Par sa politique de régulation de l'Internet totalement imposée par l'Etat, la Chine se place dans le groupe de pays totalitaires comme le Vietnam ou la Birmanie.

Depuis 2000, l'inflation du nombre de textes juridiques relatifs à la réglementation et régulation d'Internet démontre une attention accrue pour ce médium.

2.1. Les acteurs de la régulation

Une douzaine d'entités au moins ont autorité sur les réseaux et les contenus. Parmi les principales, citons le Ministère de la Sécurité Publique (MSP), responsable de la régulation générale de l'accès à Internet depuis 1994, de la sécurité nationale des systèmes informatisés, de la prévention et du contrôle des virus et données dangereuses. Il accorde également les licences pour les produits de sécurité qui doivent être évalués par le centre de tests du MSP situé à Tianjin [8]. Il a donc là un rôle important, en mesure de maîtriser les relations avec les compagnies chinoises mais aussi étrangères qui vendent de tels outils. Le Ministère de la Sécurité d'Etat (MSE) a pour mission claire de lutter contre les tentatives des pays étrangers de porter atteinte à la sécurité de l'information du pays. Le Ministère de l'Industrie de l'Information (MII), qui supervise l'infrastructure Internet, a pour sa part la responsabilité du contrôle et de l'attribution des licences dans le domaine des télécommunications, de l'industrie logicielle et des fournisseurs de contenus Internet. Interviennent également dans le processus de régulation, le Bureau des Secrets d'Etat, le Département de propagande central qui veille à ce que les éditeurs n'impriment que des contenus en accord avec la ligne idéologique du Parti Communiste, l'administration générale de la presse et

Daniel VENTRE
CNRS
daniel.ventre@gern-cnrs.com

des publications (AGPP) qui accorde des licences et surveille les publications (les journaux, périodiques, livres et sites internet), le Département du Commerce, l'administration d'Etat de l'Industrie et du Commerce, le Ministère du commerce extérieur et de la coopération pour les questions liées au commerce électronique, le Ministère de la Culture pour les contenus, etc.

2.2. Régulation des Contenus

La régulation des contenus, dominée par l'obsession d'un black-out total des discussions politiques, est liée à la volonté de maintenir un équilibre politique et social établi. Le gouvernement chinois a toujours maîtrisé l'information reçue par ses citoyens au travers des médias officiels. L'arrivée d'Internet a posé un défi nouveau au gouvernement chinois, qui a rapidement mis en œuvre des modalités de contrôle de l'usage de ce médium comme il avait contrôlé les autres jusqu'alors, tout en encourageant officiellement le développement des activités en ligne, conscient des enjeux économiques.

De nombreux textes établissent la liste des contenus illicites. Le « **Règlement pour la gestion, la protection, la sécurité d'Internet et des réseaux informatisés** » du 11 décembre 1997, les « Mesures concernant les services d'information sur Internet » (25 septembre 2000) interdisent l'utilisation d'Internet pour porter atteinte aux intérêts de la sécurité nationale, révéler des secrets d'Etat. Les interdictions concernent la création, reproduction, diffusion de contenus incitant à l'opposition à la Constitution, aux lois, règlements administratifs, incitant à la division du pays, portant atteinte à l'unité nationale, incitant à la discrimination entre les nationalités, déformant la réalité, propageant des rumeurs, faisant la promotion de superstitions moyenâgeuses, de contenus à caractère sexuel, proposant des jeux de pari, faisant l'apologie du meurtre, incitant au terrorisme, au crime, de nature injurieuse ou diffamatoire.

La lutte contre la subversion

La subversion consiste à porter atteinte à l'unité nationale et se place donc sur le terrain politique et idéologique. Les contenus les plus censurés, subversifs et dangereux, concernent ceux à caractère pornographique, traitant du Tibet, de Taiwan, de démocratie, ceux critiques à l'encontre du Parti, traitant du mouvement religieux Falungong déclaré secte satanique et sont interdits depuis 1999, discours prônant le terrorisme ou menaçant la sécurité et l'unité nationale, mais aussi les sites d'éducation étrangers (des universités, le MIT...), les sites sur la santé, les médias étrangers comme la BBC, les sites gouvernementaux étrangers. En 2001, quatre cyberdissidents (fondateurs du site lib126.com) sont arrêtés et condamnés (8 à 10 ans de prison) pour subversion, pour avoir diffusé des textes politiques, critiquant le gouvernement. Un internaute a été accusé [9] de subversion pour avoir téléchargé 500 textes sur des sites de démocrates chinois basés à l'étranger en 2002.

Les secrets d'Etat

Les nombreux textes de loi relatifs à la sécurité nationale et aux secrets d'Etat sont des éléments majeurs du contrôle des contenus de l'Internet en Chine. Les termes « secrets d'Etat » englobent l'information confidentielle dans des domaines allant du développement social à la technologie, aux relations internationales, à la défense nationale, à l'économie, tout ce qui peut concerner la sécurité et les intérêts de l'Etat. La définition des secrets d'Etat est cependant très large. Un simple sujet d'examen d'université a été considéré par un tribunal comme un secret d'Etat. L'information a également été déclarée secret d'Etat, notamment celle émanant de la presse étrangère, tant qu'elle n'a pas été reprise par une agence de presse officielle du gouvernement.

La responsabilité de la protection des secrets d'Etat est collective, imposée à tous les citoyens par la Constitution.

Le Bureau des Secrets d'Etat est le bras de l'appareil communiste pour contrôler tout ce qui relève du secret d'Etat. Le Bureau a souhaité étendre son pouvoir à l'Internet en étendant la Loi sur les Secrets d'Etat à l'Internet, en 2000. Ce texte interdit explicitement la connexion aux réseaux (Internet, tout réseau d'information public), directement ou indirectement, de toute machine contenant des informations secrets d'Etat. L'interdiction s'applique bien sûr aux *chat rooms*, *newsgroups*... dont les opérateurs sont responsables des contenus à cet égard. S'ils trouvent des contenus suspects, ils doivent les dénoncer. Les personnes reconnues coupables d'avoir fourni des secrets d'Etat à des étrangers via l'Internet sont passibles de la peine de mort dans les cas les plus graves. Quand le cas est simplement « sérieux », la peine peut être de 10 années de prison (**article 111 de la loi pénale de Mars 1997**).

La liberté de parole

La liberté de parole est inscrite dans la **Constitution** de 1982. **L'article 35** précise que les citoyens jouissent de la liberté de parole, de presse, de réunion, d'association... Les limites à la liberté d'expression apparaissent dans l'interdiction de publier ou diffuser des critiques des responsables du parti ou des opinions qui remettent directement en cause l'autorité politique du Parti Communiste Chinois.

* 2 septembre 1999, province de Hebei, un internaute est arrêté, condamné à 4 ans de prison pour avoir envoyé via Internet des messages à caractère antigouvernemental. Août 2001, arrestation à Tianshui, province de Gansu, d'un ancien policier, et condamnation à 11 ans de prison pour téléchargement et impression de 50 articles réactionnaires sur Internet.

* En 1997 deux nouveaux crimes ont été définis : les crimes informatiques et la collusion avec les organisations étrangères ou les individus dont l'objet est de déstabiliser le régime socialiste en faisant courir des rumeurs ou par tout autre moyen. En

application de ce nouveau texte, un ingénieur logiciel de Shanghai a été condamné en 1998 pour tentative de déstabilisation du gouvernement par l'envoi à 30 000 adresses e-mail en Chine, d'un texte pointant vers un magazine électronique pro-démocratique hébergé aux États-Unis.

Diffamation

Des principes généraux de droit des citoyens au respect de leur dignité sont inscrits dans la **Constitution (art. 38).** Le droit à la réputation est inscrit dans les principes généraux de la **Loi Civile** [10] **de la République Populaire de Chine** (1986) ou encore dans la **Loi Pénale** de mars 1997 qui condamne la diffamation de 5 ans de prison **[11]**.

D'autres textes spécifiques à l'internet comme les « Mesures administratives pour les services d'information sur Internet » (2000) interdisent aux fournisseurs de services de reproduire, produire, diffuser de l'information contenant des insultes ou des propos diffamants.

La diffamation n'est pas clairement définie dans les textes. Les tribunaux doivent donc interpréter. Les réparations de la diffamation sont civiles : réparation des dommages, compensation des pertes... Il peut être demandé par les tribunaux que les coupables fassent de plus des excuses publiques, signent un acte de repentance. Des amendes et peines de prison peuvent être toutefois ordonnées. Le premier cas de diffamation en ligne a été enregistré en 1999 à Pékin opposant un fabricant d'ordinateurs à un client mécontent ayant exprimé son sentiment sur un site et deux magazines ayant repris ses propos. Le fabricant obtient gain de cause, ayant enregistré de fortes pertes commerciales du fait de l'atteinte à son image de marque. En juin 2001, une autre affaire voit la mise en accusation d'un hébergeur de BBS **[12]**. Sur requête du plaignant, les contenus diffamants ont été retirés. La responsabilité de l'hébergeur n'a pas été retenue **[13]**. Quant à l'identité des auteurs, elle n'a pu être fournie, la plainte intervenant au-delà du délai légal de conservation des données fixé à 60 jours.

Informations nominatives et protection de la vie privée

En comparaison avec les pays occidentaux qui considèrent les droits de l'individu et le respect de la vie privée comme fondamentaux, la Chine s'est peu intéressée à ces concepts et à leur application dans la gestion des systèmes d'information.

Certes il n'y a pas de loi spécifique pour la protection des données à caractère personnel, comme en France la loi informatique et libertés. Mais bien que rares, les textes relatifs à la protection des données personnelles ou au respect de la vie privée ne sont pas totalement inexistants.

- Une protection est accordée par la **loi sur les statistiques** (1983) qui prévoit que les données collectées lors de recherches ne peuvent être divulguées sans le consentement des personnes.

- Un **règlement** de 1998 **relatif à l'administration des réseaux** prohibe « l'invasion de la vie privée d'autrui en diffusant de fausses informations ou en utilisant le nom d'autrui pour diffuser de l'information » (ce dernier point s'applique donc à l'usurpation d'identité).

- Des **Mesures pour la sécurité et l'administration des réseaux internationaux** (1997) rappellent que la liberté de communication et le secret des communications des utilisateurs sont protégés par la loi.

- La **Constitution** reconnaît un droit à la dignité (**article 38**), le droit au respect des correspondances privées (**article 40**), principe que l'on retrouve dans la loi criminelle de 1979.

- Il est interdit d'intercepter, modifier ou détruire les courriers électroniques d'autrui (Décision relative à la sécurité de l'Internet, article 4.2).

- L'information personnelle des utilisateurs est protégée contre la divulgation publique non autorisée par les fournisseurs de services de messagerie électronique.

- Le « **Règlement pour le Courtage en ligne** » impose le cryptage des données clients quand elles transitent par Internet.

- Des réglementations édictées au niveau local, notamment à Shenzhen (2002) et Shanghai (2003) prennent en compte les droits des consommateurs. Le texte de Shanghai interdit la divulgation sans autorisation des consommateurs de leurs données nom, genre, profession, niveau d'éducation, adresse, situation familiale, maladies. Ces initiatives sauraient-elles être généralisées à la Chine toute entière ?

Mais l'Etat possède tout pouvoir et malgré la **loi de procédure criminelle** (article 116) qui n'autorise l'Etat à accéder aux e-mails ou télécommunications privées que dans des cas bien définis (et théoriques) d'enquêtes et de procédures criminelles, les fournisseurs d'accès et de contenus doivent fournir les informations personnelles des utilisateurs qui enfreignent les lois ou postent des contenus interdits. La presse internationale accusait récemment Yahoo ! d'avoir fourni aux autorités chinoises des informations permettant l'arrestation d'un internaute chinois et sa condamnation à 10 ans de prison en avril 2005. Dans cette affaire Yahoo ! a fourni des détails concernant les communications par mail de cet internaute et permis aux autorités chinoises d'établir qu'il avait divulgué des secrets d'Etat. Yahoo ! se défend en disant avoir simplement respecté la législation et les obligations du pays d'exercice, comme il le fait partout dans le monde.

Si les textes existent pour préserver la vie privée, ce n'est que très récemment que cette notion a fait son apparition, en particulier au travers des affaires de diffamation concernant des personnalités du spectacle et du sport. L'intrusion des médias dans la vie privée concerne essentiellement ces personnages publics, car elle prend la forme d'atteinte à la réputation.

Cybercriminalité

Selon la **loi pénale** du 14 mars 1997, est qualifié de cybercrime tout acte criminel ayant pour cible les systèmes informatisés. Les crimes dans lesquels l'ordinateur, les réseaux ne sont qu'un outil, ne sont pas considérés comme cybercrimes, mais comme crimes ordinaires, relevant d'autres formes : atteinte à la sécurité de l'Etat, diffusion de contenus pornographiques...

Les premiers actes de cybercriminalité sont enregistrés en Chine dans le milieu des années 1980. La première victime est le système

bancaire. Jusqu'au milieu des années 1990, la cybercriminalité reste simple dans sa forme (essentiellement crimes contre la propriété). Il y a peu de cas, mais ils ont un impact fort. L'attention du public est attirée sur ce nouveau phénomène avec l'apparition de virus [14], notamment le virus « ping pong », qui laisse les utilisateurs non préparés totalement désarmés. En 1989 apparaissent les virus de propagande politique. Le gouvernement réagit alors, car le virus est entré sur le champ politique, domaine sensible. Le cybercrime est dès lors considéré comme un crime à part entière.

La deuxième période de la cybercriminalité commence en 1996 avec l'extension massive et la croissance exponentielle des réseaux et du nombre d'utilisateurs, même si l'internet chinois connaît un développement relativement tardif en Chine (ce n'est qu'en 1987 que le premier mail chinois est envoyé hors du pays, et le premier réseau connecté à l'extérieur est le réseau CANET – *China Academic Network* – en 1988).

En décembre 2000, est promulguée une loi qualifiant de « cybercrime » et « cyberdissidence » la propagation de rumeurs, la diffamation, l'incitation à la déstabilisation du gouvernement et du système socialiste ou à la division du pays.

Le piratage de systèmes est réprimé sévèrement. En 1999, un citoyen chinois reconnu coupable d'avoir volé de fortes sommes d'argent à une grande banque (*Industrial and Commercial Bank of China*) en s'introduisant dans les systèmes informatisés de l'entreprise, a été condamné à mort. Les cas de piratage ne sont pas isolés malgré la menace des sanctions. Mars 2005, arrestation en Chine d'un individu soupçonné du piratage de 100 000 ordinateurs afin de les inclure dans un réseau de PC zombies, capables de lancer des attaques DDoS. 60% des ordinateurs touchés étaient chinois, dont certains appartenaient à l'administration de l'Etat.

Mais quand les pirates prennent (de plus en plus) pour cible des réseaux à l'étranger, le gouvernement par son silence et le manque de réaction de ses autorités, semble cautionner ces actes. Nombreuses sont déjà les accusations portées à l'encontre de la Chine par des entreprises ou organismes d'Etat dans le monde, sans qu'il soit toujours possible de démontrer avec exactitude l'origine des attaques de type intrusion dans les systèmes, défiguration de sites, altération de données, vol de données, DoS, *flooding*, propagation de virus…

Les pirates chinois, s'érigeant en chevaliers défenseurs de leur nation, s'en prennent à diverses cibles, parmi lesquelles les États-Unis [15] (désigné comme le rival stratégique et idéologique majeur), Taiwan, la Corée du Sud sont des victimes privilégiées, mais aussi le Royaume-Uni, le Canada, l'Australie, le Japon.

Des attaques à l'aide de *trojans* visant des entreprises occidentales ont été enregistrées ces derniers mois. En 2004, le trojan Myfip fait son apparition. Le *reverse engineering* du trojan révèle qu'il envoie des données dérobées à ses victimes, des entreprises, à un internaute localisé à Tianjin, 3ème ville de Chine, 2ème *hub* de l'industrie en particulier électronique. La source du trojan Netthief, qui a attaqué des entreprises au Royaume Uni, a également été localisée en Chine. Les trojans s'en prennent aux données sensibles : PI, secrets commerciaux… La Chine continentale est montrée du doigt. Taiwan n'est pas en reste [16]. Rien ne prouve bien entendu que les autorités chinoises sponsorisent directement ces activités. Mais en ne les réprimant pas, elles les cautionnent. Il apparaît que les autorités ne répriment pas ce qui ne va pas à l'encontre des intérêts de l'Etat chinois, même si la manière est illégale. Quel recours de la part des victimes ?

2.3. La régulation des Acteurs

Un vaste système de licences et d'autorisations est instauré pour les utilisateurs et pour les fournisseurs d'accès, de services, de contenus. Responsabilisation des acteurs et criminalisation des actes déviants sur l'Internet en conformité avec le code pénal qui condamne les « crimes [17] mettant en danger la sécurité nationale » gouvernent la régulation des systèmes d'information chinois.

Responsabilisation des fournisseurs

Le système de responsabilisation des divers fournisseurs est construit de sorte que pour se protéger, ces derniers n'ont d'autre solution que d'instaurer eux-mêmes des systèmes de contrôle, censure et filtrage. Le concept de coresponsabilité est décliné dans de nombreux textes juridiques relatifs au contrôle de l'Internet. Les auteurs ne sont pas seuls responsables. Le hébergeurs, responsables de newsgroups, chat rooms, BBS… doivent implémenter leurs propres mécanismes de contrôle afin de contribuer à la surveillance des contenus diffusés, en vue de protéger les secrets d'Etat ou de ne pas reproduire de contenus illicites (subversion, diffamation, pornographie…). Les fournisseurs de contenus doivent conserver copie de tous les contenus rendus accessibles [18]. **L'arrêté sur la gestion des contenus sur l'Internet** du 25 septembre 2000 leur interdit la diffusion d'informations dommageables et malsaines.

Les fournisseurs d'accès doivent enregistrer et conserver les données concernant leurs clients : numéro de compte, de téléphone, adresse IP, conserver copie des données enregistrées durant 60 jours et les fournir aux autorités dans le cadre d'enquêtes.

Le concept de coresponsabilité induit un système de responsabilités en cascade : si le coupable (l'auteur) n'est pas identifié, la justice peut se retourner sur un autre maillon : l'hébergeur, le responsable du site, etc. L'hébergeur qui n'est pas en mesure de respecter ces contraintes risque le retrait de sa licence et l'arrestation de ses personnels. C'est en raison de l'existence de ce principe de coresponsabilité qu'en août 2002 a été proposé et signé par plus de 300 hébergeurs, portails, entreprises, universités, le « **pacte public d'auto-discipline de l'industrie de l'internet en Chine** ». Ce texte est un acte de contrôle volontaire des contenus. Les signataires promettent de purger les contenus de tout ce qui est répréhensible au regard de la loi (pornographie, atteinte au gouvernement, subversion…). MSN Spaces, service de blogs, bloque des termes « interdits » (démocratie, indépendance de Taiwan…)

Depuis juin 2005, les sites chinois doivent obligatoirement afficher sur leur page d'accueil leur numéro d'identification. Ceux qui ne le feraient pas seront progressivement fermés. Ces numéros d'identification sont ceux attribués lors de l'enregistrement/déclaration obligatoire auprès des autorités. Lors de cette déclaration, le nom et les coordonnées du responsable du site sont fournis. Pour bloquer l'accès aux sites qui n'ont pas été

enregistrés auprès des autorités au 1er juin 2005, un nouveau système de gestion des contenus Internet, dénommé « *night crawler* », a été mis au point. Le système ne s'en prend qu'aux sites ayant une adresse IP attribuée en Chine.

Les fournisseurs de services d'informations en ligne (presse) sont fortement contraints par la classification en « secret d'Etat » de l'information. Les sites web non licenciés ne peuvent pour leur part que publier l'information déjà fournie par d'autres médias officiels. La loi restreint la diffusion d'information provenant des médias étrangers, considérée comme « secret d'Etat » tant qu'elle n'a pas fait l'objet d'une publication par un organe de presse officiel.

Responsabilisation collective : l'exemple de la lutte contre la pornographie

La diffusion de pornographie, qu'elle soit sous forme d'images, littérature, vidéo est passible de deux ans de prison [19], et plus quand le public est mineur.

Dans la droite lignée de son approche prohibitionniste de la prostitution et de la pornographie, en vigueur depuis l'arrivée des communistes au pouvoir en 1949, le gouvernement s'est lancé le 16 juillet 2004 dans une campagne de répression de la pornographie sur le web impliquant 14 ministères. Justification de cette campagne anti-pornographie sur Internet : l'impact sur la société et en particulier les mineurs. Le gouvernement implique la population dans ses actions. La police rémunère les informateurs qui signalent aux forces de police les sites web pornographiques (de l'ordre de 50 à 200 €).

Toutes les entreprises du secteur de l'internet et des télécoms sont invitées à participer à cette lutte. China Unicom, China Mobile se sont vues demander de contrôler les messages SMS. Les banques sont invitées à surveiller les transactions financières en ligne. Durant les 4 premiers mois de la campagne anti-pornographie en 2004, 1125 sites ont été fermés et 445 personnes arrêtées [20], certaines condamnées à plusieurs années de prison et des amendes élevées. Le premier jugement a eu lieu en août 2004.

Une femme a été condamnée à 4 années de prison pour avoir proposé des strip-teases sur un site à accès payant. Le jugement a mis en avant la sévérité des peines encourues, mais l'impact de ces mesures répressives périodiques dont la Chine semble coutumière [21] ne peut donc être que limité, voire contraire à l'effet recherché. La révélation des gains potentiels de ces activités pourrait créer des vocations. L'Etat lui-même semble tenté. Le site du Henan Baoye, maintenu par un journal du Parti Communiste Chinois, le Henan Daily [22], a utilisé des contenus à caractère pornographique (des forums à caractère sexuel) pour accroître la fréquentation.

Les cybercafés

Ces établissements qui n'ont de bar que le nom, puisqu'il s'agit de salles d'ordinateurs, attirent un grand nombre d'internautes, une majorité de jeunes, et exercent souvent dans l'illégalité, sans la moindre licence. Périodiquement, la Chine durcit la législation et le contrôle des cybercafés. En 2001, une vaste enquête de 3 mois entraîne la fermeture de plus de 8000 cybercafés. La police a imposé l'installation du logiciel de filtrage de l'information « *Internet Police 110* [23] » pour bloquer les contenus à caractère pornographique et l'information subversive. La régulation et la surveillance de ces établissements se sont durcies en 2002, suite à l'incendie d'un cybercafé à Pékin faisant 24 victimes, toutes étudiantes. Toutes les plus grandes villes de Chine prirent alors des mesures pour renforcer les contrôles de ces établissements. Sous prétexte de garantir la sécurité des clients, 12000 cybercafés furent fermés entre octobre et décembre 2004 [24], visant en priorité ceux situés près des écoles primaires, des collèges, des lycées.

Les obligations pesant sur ces établissements sont nombreuses : obtenir une licence pour exercer, interdiction d'accueillir les mineurs, installer des logiciels bloquant l'accès aux contenus illicites, contrôler l'identité des clients, conserver les données de connexion pendant 60 jours permettant de lier un utilisateur à une machine et aux pages visitées, déconnecter les utilisateurs qui accèdent à des contenus illicites, en référer au département de la Culture local, pouvoir interdire l'utilisation à des fins d'atteintes aux systèmes (virus, intrusion...), ne pas être situés à moins de 200 mètres d'établissements d'enseignement élémentaire et primaire ou d'habitations résidentielles, exercer dans des créneaux horaires définis... Les données de connexion doivent pouvoir ensuite être fournies aux autorités à leur demande. Aux outils logiciels de contrôle s'ajouteraient des moyens policiers humains, les Big Mamas (super inquisiteurs) chargés de surveiller les internautes dans les espaces publics.

2.4. Régulation par la technologie

Faisant suite au projet « Grande muraille électronique », visant à séparer l'Internet chinois du reste du monde par un *firewall* massif, le projet « Bouclier Doré » initié en 2000 vise à contrôler l'usage des technologies de l'information au moyen d'un système de surveillance massive, à mettre en œuvre de gigantesques bases de données d'informations nominatives, à utiliser de multiples technologies (données, vidéos, reconnaissance vocale, biométrie...), à améliorer les capacités de réaction des forces de police dans la lutte contre le crime.

Le logiciel de filtrage « *Filter King* » a été conçu dans le cadre de ce projet et testé en 2001 dans la province de Xi'an. Surveillance et filtrage des contenus s'appuient sur des technologies similaires au système américain Carnivore [25], avec l'usage de boîtes noires qui interprètent toutes les transmissions, pouvant être accolées au serveur des FAI.

Le filtrage des contenus a lieu à plusieurs niveaux : au niveau des backbones des réseaux de la Chine et au niveau des fournisseurs qui implémentent également leurs outils de filtrage. La majorité des grands moteurs de recherche chinois filtrent les contenus par mots clefs [26] et retirent certains résultats de leurs listes. Un moteur de recherche conforme aux critères chinois a été développé par l'entreprise chinoise Sinobet et le très officiel Centre Chinois de l'Information Internet.

Le moteur, qui élimine les sites subversifs ou pornographiques, a été adopté en 2003 par sina.com.cn et par 200 autres sites chinois depuis. Des rapports font état du filtrage par la Chine de Skype, de MSN... En 2004, des *hackers* ont trouvé, associé au logiciel QQ *Instant Messaging* [27] qui est parmi les outils les plus populaires utilisés par les internautes chinois, un fichier

automatiquement installé sur la machine de l'utilisateur, nommé `COMToolKit.dll` contenant une liste de termes [28] servant au filtrage des contenus.

2.5. Résistances

Des résistances à la régulation viennent interférer dans ce processus complexe sous emprise de l'Etat.

* La « **Déclaration des droits des internautes chinois** » initiée en juillet 2002 par 18 intellectuels chinois appelle à la liberté d'expression sur Internet. Cette liberté d'expression, nombreux sont les internautes à la revendiquer.

A l'image de *Furong Jiejie* (*sister Furong*), jeune femme devenue « star » du Net après s'être autoproclamée séduisante et talentueuse et avoir diffusé sur des blogs ses photographies dans des poses suggestives. Sa popularité est remontée jusqu'aux autorités communistes qui ont fait interdire ses publications sur le net. Mu Zimei, jeune femme de 25 ans, a publié sur le net ses aventures sexuelles. Le gouvernement est intervenu pour faire cesser ces publications. Les « 9 commentaires sur le Parti Communiste Chinois », série d'essais d'auteurs anonymes sur l'histoire du parti, ont été publiés en ligne, entraînant une vague de résiliations d'adhésions au Parti. Plusieurs millions de chinois auraient publiquement renoncé à leur appartenance au Parti depuis que le site « *Tuidang* » (Quitter le PC) a été ouvert en décembre 2004 pour recenser ces résiliations.

* Les internautes chinois ont appris à contourner les filtres, en utilisant les serveurs proxy ou des logiciels tels « *Roaming Without Borders* », « *Ultrasoft* » ou autres « *Freenet* » (P2P), « *TriangleBoy* » qui permettent aux membres d'échanger des informations de manière anonyme.

Conclusion

Dans ce cadre de régulation et de réglementation sous contrôle de l'Etat, le marché chinois reste fermé à des produits étrangers, comme ceux de cryptographie, sous contrôle strict. Cependant, le développement des relations internationales dans le domaine des NTIC en Chine a été rendu possible par l'adaptation du droit, en mesure d'assurer notamment la protection des intérêts des entreprises étrangères sur le territoire chinois. La Chine, qui possède une réglementation complète en matière de propriété intellectuelle, a adhéré à de nombreuses conventions internationales relatives à la propriété intellectuelle (convention de Paris, arrangements de Madrid, accords TRIP) qui permettent aux entreprises d'obtenir réparation dans le cas de contrefaçon. La Chine, depuis son accession à l'OMC, s'est engagée à renforcer ses moyens de lutte dans ce domaine. Des procédures récentes à l'initiative de grandes compagnies de l'industrie audiovisuelle (Sony Music, Universal Music, Warner Music) s'en sont prises à la contrefaçon, attaquant les sites qui diffusent illégalement des fichiers piratés. Les titulaires de marques peuvent aussi porter plainte auprès de l'ICANN contre les sociétés squattant leurs noms de domaine. Les conflits qui se règlent via des procédures civiles et permettent d'obtenir gain de cause ne sont pas rares (IKEA, Dupont, Procter & Gamble, etc.). Mais si les entreprises étrangères peuvent faire valoir leurs droits sur le terrain de la contrefaçon, il sera sans doute plus difficile à une entreprise ou une administration étrangère, voire un simple internaute, qui serait victime de pirates chinois, de virus ou d'attaques en règle, de faire valoir ses droits.

Notes et liens

[1] http://www.netcost-security.fr, « The approaching Chinese Cyber Storm », 21 juillet 2005.

[2] Ministère de l'Industrie de l'Information, avril 2005.

[3] Première connexion à Internet en 1993.

[4] A l'été 2005, la Chine a acheté 200 routeurs à CISCO Systems.

[5] Représentant 75% du marché chinois.

[6] Alors que le système juridique taiwanais par exemple est d'inspiration allemande, tout comme celui de la Corée du Sud.

[7] Comme par exemple celles de 1983, 1996, 2001, 2003 connues sous le nom de « Yanda » (frapper fort) fixant à la police des objectifs précis en termes de résultats : mandats d'arrêt, arrestations, jugements, résultats rapides (« Les quatre rapides »). Durant l'opération « Tempête de printemps » du 23 au 25 avril 2001, la police de Hunan a « résolu » 3000 cas en deux jours et la police de Shichuan 6 704 affaires entre le 19 et le 24 avril 2001, arrêtant 19 446 personnes. Du 10 avril au 25 mai 2001, les juges de Shandong ont jugé 65 criminels par jour.

[8] Loi sur la sécurité des réseaux, 1997.

[9] En vertu de l'article 105 du Code Pénal.

[10] Chapitre 5, article 101. Section II, chapitre 6, article 120 (droit au nom, à l'image, à la réputation, à l'honneur).

[11] Article 105.

[12] Bulletin Board Service.

[13] En vertu d'un règlement pour l'administration des BBS de 2000.

[14] L'introduction délibérée de virus dans les systèmes est sanctionnée, notamment au titre de l'article 23 du « Règlement pour la protection des systèmes informatisés » de février 1996.

[15] Les réseaux du département de la défense américain seraient également victimes depuis deux ans d'attaques de plus en plus intensives. Ces incidents ont été baptisés « Titan Rain » par les agences de sécurité américaines.

[16] Selon un rapport du département de la défense américain « Taiwan Strait Posture Status », Taiwan serait le leader mondial dans le développement de techniques anti-virus, mais aussi à l'origine d'un grand nombre de virus répandus ces dernières années (parmi les plus connus en 1990 Bloody of 6/4, en 1992 Michelangelo, en 1998 Chernobyl...).

[17] Article 10 de la loi pénale du 1° juillet 1979 : « un crime est un acte qui met en danger la souveraineté et l'intégrité territoriale de l'Etat ».

[18] Texte du 25 septembre 2000. Mesures pour l'administration des Services d'Information sur Internet.

[19] Articles 169 et 170 de la loi pénale du 1° juillet 1979, article 11 des « Règlements provisoires concernant la gestion réseaux informatisés internationaux » (1997).

[20] Source : Sumner Lemon, IDG News Service, 10/11/04

[21] En 1989 et 1990, une campagne de lutte contre les « six démons », dont fait partie la prostitution, a entraîné l'arrestation de 243 183 personnes liées à la prostitution.

[22] Source : Epoch Times. 3 septembre 2005.

[23] Numéro d'appel d'urgence de la police en Chine.

[24] Xinhua Economic News Service, 1er novembre 2004.

[25] Système du FBI pour surveiller les courriers électroniques et autres trafics via les ISP.

[26] Voir liste de termes censurés : www.opennetinitiative.net/bulletins/008/bbs.pdf ou www.zonaeuropa.com/20040902_1.htm

[27] Développé par la société Tencent.

[28] Termes à caractère sexuel, religieux, politique, noms de personnalités politiques, termes génériques comme « vérité », « idée »...

CVSS (Common Vulnerability Scoring System) : un système en devenir ?

Depuis un certain nombre d'années, de nombreuses organisations et de nombreux constructeurs ont établi des systèmes d'évaluation des vulnérabilités des composants des Systèmes d'Information. Malheureusement, jusqu'à maintenant, aucun accord et aucune norme n'avaient été établis pour harmoniser ces différents systèmes. C'est désormais chose faite avec le CVSS : Common Vulnerability Scoring System. Sponsorisée par de grands acteurs comme Cisco, Symantec, Microsoft, ISS ou encore le CERT américain, cette initiative semble partie sur de bonnes bases... avant une adoption définitive dans un futur proche ?

Présentée de façon officielle lors de la dernière conférence RSA (14-18 février 2005), ce système constitue une nouvelle nomenclature qui vise à évaluer la criticité des menaces informatiques. Il doit notamment permettre de fournir aux clients de tout logiciel informatique une notation rapide des vulnérabilités les affectant sur la base de formules mathématiques. Soutenue par des géants du secteur informatique (Microsoft, Cisco, Symantec...), le département américain en charge de la sécurité du territoire et des organisations telles que eBay, Qualys et Mitre, cette classification se veut être une référence universelle.

1. Définition des différents indicateurs

Avant de définir à proprement dit les différents indicateurs mis en place par le CVSS, il convient de définir le périmètre qu'il adresse, c'est-à-dire de préciser ce qu'on entend par « vulnérabilité ». Au sens où l'entendent les concepteurs du CVSS, une vulnérabilité est : « un ensemble de conditions qui amène ou peut amener à une réduction implicite ou explicite de la confidentialité, de l'intégrité ou de la disponibilité du système d'information. Des exemples d'effets non autorisés ou non attendus de la vulnérabilité peuvent inclure les événements suivants :

- Usurpation d'un compte pour exécution de commandes ;
- Escalade des permissions relatives aux contrôles d'accès ;
- Réalisation d'un déni de service ;
- Suppression de données sans permission ;
- Exploitation de vulnérabilités dans l'implémentation de mécanismes de chiffrement afin de réduire le temps de recouvrement de données chiffrées... »

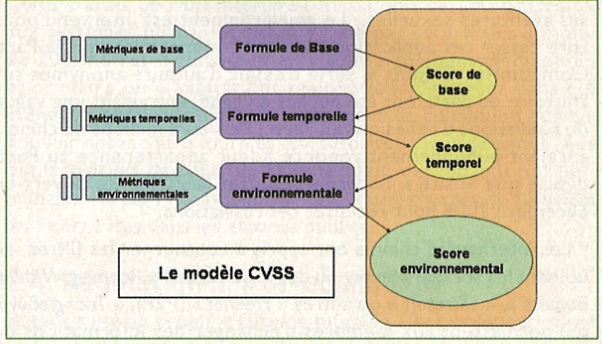
CVSS est un système modulaire composé de trois groupes distincts qui combinent les caractéristiques intrinsèques de la vulnérabilité. Chacune des métriques est évaluée d'une façon spécifique et chaque groupe est calculé par une combinaison pondérée des métriques associées. Une métrique est une caractéristique de la vulnérabilité qui peut être quantitativement ou qualitativement mesurée. Ces valeurs sont alors regroupées dans trois groupes : base, temporel et environnemental.

a. Le groupe de base contient toutes les métriques intrinsèques et fondamentales à la vulnérabilité qui ne changent pas au cours du temps et en fonction des environnements.

b. Le groupe temporel contient les métriques de la vulnérabilité qui sont dépendantes du temps et évoluent avec l'ancienneté de la vulnérabilité

c. Le groupe environnemental contient les caractéristiques des vulnérabilités qui sont dépendantes de leur implémentation et qui sont spécifiques à chaque environnement utilisateur.

Figure 1



Dans la suite de cet article, nous allons aborder en détail les différentes métriques de chaque groupe.

1.1 Indicateur de base

Suite à la découverte d'une vulnérabilité, cette dernière aura des aspects qui ne vont jamais changer. Ces caractéristiques immuables n'évolueront pas avec le temps, ni d'un environnement à l'autre. L'indicateur de base capture les métriques d'accès et d'impact. Les métriques d'accès sont au nombre de trois et elles caractérisent les moyens d'accès au système vulnérable avec la prise en compte des éventuels mécanismes de restrictions d'accès. Il y a également trois métriques, qui concernent les notions traditionnelles de confidentialité, d'intégrité et de disponibilité (**Tab. 1**).

1.2 Indicateur de temps

Au cours du temps, la qualification d'une vulnérabilité va évoluer : certaines caractéristiques vont changer. Avec le temps, les informations sur la disponibilité d'un correctif de sécurité et le nombre d'exploitation de la vulnérabilité vont probablement évoluer. Les métriques temporelles de CVSS capturent les caractéristiques d'une vulnérabilité qui changent avec le temps (**Tab. 2**).

1.3 Indicateur environnemental

L'impact d'une vulnérabilité peut varier de façon importante d'un environnement à l'autre. Les métriques environnementales de CVSS capturent les caractéristiques des vulnérabilités qui sont liées à l'environnement réseau et à la distribution des systèmes (**Tab. 3**).

Sylvain Roger
sylvain.roger@solucom.fr
Consultant Sécurité des Systèmes d'Information/SoluCom,
http://www.solucom.fr

Métrique	Définition	Valeur possible	Poids
Vecteur d'accès	Précise si la vulnérabilité est exploitable localement ou à distance.	Local Distant	Local : 0.7 Distant : 1.0
Complexité d'accès	Précise la complexité de l'attaque nécessaire à mettre en place pour exploiter la vulnérabilité une fois que le système cible a été atteint. Par exemple l'attaque pourra être jugée élevée si le système n'est exploitable qu'avec une interaction de la victime.	Faible Élevé	Faible : 1.0 Élevé : 0.8
Authentification	Précise si l'exploitation de la vulnérabilité nécessite d'être authentifié vis-à-vis du système. Le type de mécanisme d'authentification ne rentre pas en ligne de compte.	Nécessaire Pas nécessaire	Nécessaire : 0.6 Pas nécessaire : 1.0
Impact sur confidentialité	Précise l'impact sur la confidentialité des informations du SI. Doit être considéré comme partiel si l'attaquant n'a pas le choix de ce qui est découvert, autrement complet.	Aucune Partiel Complet	Aucune : 0 Partiel : 0.7 Complet : 1.0
Impact sur intégrité	Précise l'impact sur l'intégrité des informations du SI. Doit être considéré comme partiel si l'attaquant n'a pas le choix de ce qui est modifiable, autrement complet.	Aucune Partiel Complet	Aucune : 0 Partiel : 0.7 Complet : 1.0
Impact sur disponibilité	Précise l'impact sur la disponibilité des informations du SI. Doit être considéré comme partiel si interruptions ou ralentissement, complet si arrêt total.	Aucune Partiel Complet	Aucune : 0 Partiel : 0.7 Complet : 1.0
Répartition Impact	Permet de donner un poids plus important à un des trois impacts décrits précédemment.	Normal Confidentialité Intégrité Disponibilité	Normal : 0.333 Confidentialité : 0.5 Intégrité : 0.25 Disponibilité : 0.25

Tableau 1 : Les métriques de l'indicateur de base

Métrique	Définition	Valeur possible	Poids
Exploitabilité	Précise la complexité du processus d'exploitation de la vulnérabilité. Au fur et à mesure, le code d'exploitation peut être publié et être diffusé.	Non prouvée Preuve de faisabilité Fonctionnelle Élevée	Non prouvée : 0.85 Preuve de faisabilité : 0.9 Fonctionnelle : 0.95 Élevée : 1.0
Niveau de correction	Précise le niveau de correctif disponible à la date de l'évaluation de la vulnérabilité.	Correctif officiel Correctif temporaire Contournement Indisponible	Correctif officiel : 0.87 Correctif temporaire : 0.90 Contournement : 0.95 Indisponible : 1.00
Confiance du rapport	Mesure le degré de confiance en l'existence de la vulnérabilité et la crédibilité de sa description	Non confirmé Non corroboré Confirmé	Non confirmé : 0.9 Non corroboré : 0.95 Confirmé : 1.00

Tableau 2 : Les métriques de l'indicateur de temps

Métrique	Définition	Valeur possible	Score
Dégâts collatéraux potentiels	Les dommages collatéraux peuvent inclure des dommages financiers, physiques, atteinte à la réputation ...	Aucune Faible Moyenne Élevée	Aucune : 0 Faible : 0.1 Moyenne : 0.3 Élevée : 0.5
Distribution de la cible	Indicateur spécifique à l'environnement qui donne une estimation du pourcentage de systèmes au sein de l'environnement qui peuvent être affectés par la vulnérabilité	Aucune Faible Moyenne Élevée	Aucune : 0 Faible : 0.25 Moyenne : 0.75 Élevée : 1.00

Tableau 3 : Les métriques de l'indicateur environnemental

2. Un score final par vulnérabilité

Le score final d'une vulnérabilité n'est finalement valable qu'à un moment donné et pour un environnement donné, car il inclut les trois indicateurs : base, temporel et environnemental. Il est toujours possible de distinguer les trois scores séparément néanmoins. L'établissement du score final se fait donc en trois étapes, une pour chaque indicateur. Chaque indicateur a sa propre formule mathématique qui consiste en l'établissement de pondération entre les différentes métriques.

a. Score issu des métriques de base : Les métriques ayant le poids le plus important sont les métriques d'impact. Elles qualifient l'effet global que pourrait avoir l'exploitation de la vulnérabilité sur des systèmes cibles. Il se calcule de la façon suivante :

```
Score de base = 10 x (Vecteur d'accès) x (Complexité d'accès) x (Authentification)
x ((Impact Confidentialité) x (Répartition Impact) + (Impact Intégrité) x
(Répartition Impact) + (Impact Disponibilité) x (Répartition Impact)) où chaque
métrique peut prendre les valeurs décrites précédemment.
```

b. Score issu des métriques de base et temporelles : Le score temporel ajuste le score de base en prenant en compte les facteurs qui peuvent changer avec le temps. Il sera toujours inférieur ou égal au score de base avec un

facteur réducteur de 25% au maximum. Il se calcule de la façon suivante :

Score temporel = Score de base x (Exploitabilité) x (Niveau de correction) x (Confiance du rapport) où chaque métrique peut prendre les valeurs décrites précédemment.

c. Score issu des métriques de base, temporelles et environnementales : Le score environnemental se sert du score temporel comme base et prend en compte les aspects liés à l'environnement de l'organisation. Le score environnemental peut être plus élevé ou plus faible que le score temporel. Il se calcule de la façon suivante :

Score Environnemental = (Score Temporel + (10 - Score Temporel) x (Dégâts collatéraux potentiels) x (Niveau de distribution de la cible) où chaque métrique peut prendre les valeurs décrites précédemment.

Un fichier reprenant l'ensemble de ces scores est mis à votre disposition à l'adresse : **http://www.miscmag.com/MISC22/CVSS/cvss.xls**

3. Deux exemples d'application

Maintenant que nous avons décrit de façon précise ce système, passons à son application par quelques exemples très précis. Il est à noter que les scores obtenus pour les métriques temporelles et environnementales sont bien évidemment très subjectives.

3.1 Exemple n°1 : La faille LSASS de Microsoft

Prenons l'exemple de la vulnérabilité LSASS de Microsoft (CAN-2004-0533) en calculant le score de cette vulnérabilité au 12 juillet 2005 pour un parc d'une petite entreprise comptant 200 machines dont 180 sont sous Microsoft Windows 2000 et étant correctement patchée. Dans notre exemple, nous pouvons attribuer les valeurs suivantes aux métriques du score de base :

a. Vecteur d'accès : 1 (la vulnérabilité est exploitable à distance) ;

b. Complexité d'accès : 1 (aucune interaction avec l'utilisateur n'est nécessaire) ;

c. Authentification : 1 (il n'est pas nécessaire d'être authentifié vis-à-vis du système) ;

d. Impact sur la confidentialité : 1 (l'impact sur la confidentialité est complet) ;

e. Impact sur l'intégrité : 1 (l'impact sur l'intégrité est complet) ;

f. Impact sur la disponibilité : 1 (l'impact sur la disponibilité est complet) ;

g. Répartition impact : 0, 33 (répartition normale entre les différents impacts).

D'où un score de base = 10 x 1 x 1 x 1 x (1 x 0,33 + 1 x 0,33 + 1 x 0,33) = 10

Concernant la métrique temporelle :

a. Exploitabilité : 0,95 (un code d'exploitation fonctionnelle est publiquement disponible sur Internet)

b. Niveau de correction : 0,87 (un correctif a été proposé officiellement par Microsoft)

c. Confiance du rapport : 1 (la vulnérabilité a été confirmée par l'éditeur)

D'où un score temporel = 10 x 0,95 x 0,87 x 1 = 8,3

Enfin concernant la métrique environnementale :

a. Dégâts collatéraux potentiels : 0 (les données sensibles de l'entreprise ne sont pas accessibles directement)

b. Distribution potentielle : 1 (90% du parc de l'entreprise est affectée par cette vulnérabilité)

D'où un score Environnemental = (8,3 + (10 - 8,3) x 0 x 1) = 8,3

3.2 Exemple n°2 :

Prenons l'exemple de la vulnérabilité BGP affectant l'IOS de Cisco (CAN-2004-0589) en calculant le score de cette vulnérabilité au 12 juillet 2005 pour un parc d'une grande entreprise comptant 50 routeurs exclusivement Cisco sous sa responsabilité.

Pour ce deuxième exemple, nous pouvons attribuer les valeurs suivantes aux métriques du score de base :

a. Vecteur d'accès : 1 (la vulnérabilité est exploitable à distance) ;

b. Complexité d'accès : 0,8 (la complexité d'accès pour l'exploitation de la vulnérabilité est élevée) ;

c. Authentification : 1 (il n'est pas nécessaire d'être authentifié vis-à-vis du système) ;

d. Impact sur la confidentialité : 0 (aucun impact spécifique sur la confidentialité à noter) ;

e. Impact sur l'intégrité : 0 (aucun impact spécifique sur l'intégrité à noter) ;

f. Impact sur la disponibilité : 1 (l'impact sur la disponibilité est complet) ;

g. Répartition impact : 0, 25 (seule l'impact sur la disponibilité a été clairement mis en évidence à la date d'évaluation de la vulnérabilité).

D'où un score de base = 10 x 0,8 x 1 x 1 x (0 x 0,33 + 0 x 0,33 + 1 x 0,5) = 4

Concernant la métrique temporelle :

c. Exploitabilité : 0,85 (l'exploitation n'a pas été prouvée)

d. Niveau de correction : 1 (pas de correctif proposé)

e. Confiance du rapport : 1 (la vulnérabilité a été confirmée par l'éditeur)

D'où un score temporel = 4 x 0,85 x 1 x 1 = 3,4

Enfin concernant la métrique environnementale :

c. Dégâts collatéraux potentiels : 0 (les données sensibles de l'entreprise ne sont pas accessibles directement)

d. Distribution potentielle : 1 (90% du parc de l'entreprise est affecté par cette vulnérabilité)

D'où un score Environnemental = (3,4 + (10 - 3,4) x 0 x 1) = 3,4

4. Une adoption future ?

Il manquait cependant encore un hébergeur au système CVSS pour promouvoir son usage à l'aide de portails Internet et de démonstrations en ligne comme le fait par exemple la société Mitre dans le cas de la base de données du CVE. Ce manque a été comblé, il y a peu, avec l'annonce du choix de FIRST comme hébergeur du système. Cette volonté de normaliser intervient alors que le délai entre la publication de failles logicielles et l'apparition de codes malveillants en tirant parti a fortement

diminué. Pour les responsables sécurité, il devient alors nécessaire de définir des niveaux de priorité aux différents correctifs afin de maintenir son parc protégé. Mercredi 16 février 2005, les différents participants de la conférence RSA avaient débattu de l'intérêt de réguler ou non l'industrie informatique en termes de sécurité. Le débat, déjà évoqué en 2004 par plusieurs grandes banques américaines, envisageait la mise en place de pénalités financières pour les éditeurs responsables de failles de sécurité sévères et répétées.

4.1 Un système séduisant...

Un tel système est en effet séduisant à plus d'un titre. D'une part, il y a un réel besoin de normalisation concernant l'évaluation des failles. Ce besoin s'exprime tout d'abord au niveau de l'indicateur de base et donc pour les bases de données publiées qu'elles soient issues d'acteurs comme BID (*BugTrack*), CVE, ISS X-Force, OSVDB, Secunia, CERT,... En effet, à l'heure actuelle, il y a un manque flagrant de standardisation entre les différents acteurs. Par exemple, aucune convention de noms n'existe pour la description des vulnérabilités.

D'autres problèmes ont été soulevés dans le passé, comme le manque d'exactitude, voire d'intégrité : un certain obscurantisme entoure l'identité des responsables des mises à jour de ces bases de vulnérabilités, l'inexactitude est parfois au rendez-vous des descriptions de certaines vulnérabilités. Or, il y a en général une forte croyance dans les informations publiées sur ces bases de vulnérabilités. La mise en place de CVSS pourrait permettre via l'indicateur de base de normaliser un certain nombre de points. Ensuite, la prise en compte de variables liées à l'environnement et au temps ne fait que crédibiliser un tel système. De plus, le système laisse encore une part de jugement à la personne responsable de l'évaluation de la faille dans un environnement et à une date donnée. L'intégration de métrique, comme les dégâts collatéraux potentiels, renforce l'objectivité de l'évaluation.

4.2 ...mais qui peut laisser sceptique

Un tel système peut également laisser songeur. Par exemple, les critiques issues de la situation actuelle peuvent se retrouver dans le système CVSS : qui va pouvoir assurer l'intégrité des informations nécessaires à l'établissement du score d'une vulnérabilité ? Le système CVSS sera toujours dépendant de la qualité des informations fournies par les personnes ayant découvert la vulnérabilité. Une des principales interrogations sur ce système concerne les pondérations mises en place pour chacune des métriques. On peut se demander quelle est la justification à ces pondérations. Pourquoi mettre une pondération de 0,87 si un correctif officiel est disponible ou de 0,9 si une preuve de faisabilité de l'exploitation de la vulnérabilité a été avancée ? Pour le moment peu de justifications ont été émises par les responsables du projet sur ces points.

Une autre critique (habilement soumise par Papy) concerne la comparaison des scores entre eux et donc des vulnérabilités sous-jacentes. Le score attribué à une vulnérabilité contient finalement beaucoup d'informations et on peut considérer qu'il agglomère trop d'informations disparates. Le manque d'un référentiel pour comparer les différents scores entre eux est important : que conclure d'une vulnérabilité qui obtient un score de 7 ? Représente-t-elle un risque pour l'entreprise ? Deux vulnérabilités ayant le même score doivent-elles recevoir le même intérêt ? On pourra y répondre que c'est peut-être là la force du système : laisser encore à l'auditeur une marge de manœuvre pour apporter son expertise d'analyse tout en formalisant un référentiel commun. Enfin pour qu'un tel système se développe, il est nécessaire qu'il soit pérenne.

Or, on peut avoir des réserves quant à la pérennité du système CVSS. Il ne semble pas que le système soit arrêté. Des modifications sont encore possibles dans le système ce qui ne contribue pas pour le moment à son développement.

Conclusion

L'avènement de la prise en compte de la sécurité dans le milieu informatique a fait exploser le nombre de vulnérabilités recensées. Il suffit de voir l'évolution du nombre de failles reportées par les bases de données consacrées. Par conséquent, un système d'évaluation est nécessaire pour pouvoir réellement estimer la criticité d'une faille. Jusqu'à maintenant, les systèmes proposés ne reflétaient pas la réalité, mais CVSS commence à répondre à certaines attentes grâce notamment à l'introduction de trois métriques différentes. Son adéquation pourrait bientôt en faire un système de référence s'il commence à être adopté complètement par certains acteurs du monde de la sécurité. Dernière actualité en date, l'équivalent du ministère de l'intérieur américain vient de lancer une initiative pour la construction d'une base de vulnérabilité nationale gérée par le NIST (*National Institute of Standards and Technology*). Cette base vient en complément de l'action de veille et de réponse à incident symbolisé par le fameux US-CERT (*US Computer Emergency Readiness Team*) qui a été contacté par un des porteurs du système CVSS pour l'intégrer dans leurs bulletins d'alertes et aux premières nouvelles la rencontre a été fructueuse... De là à penser que cet événement pourrait être le déclencheur attendu pour l'avènement de CVSS, il n'y a qu'un pas qui risque d'être bientôt franchi.

Liens

→ *NIAC Vulnerability Disclosure Working group, The Common Vulnerability Scoring System*, disponible en ligne sur **http://www.vulnerabilite.com/cvss/cvss.pdf**

→FIRST, FIRST to host CVSS : **http://www.first.org/cvss/**

→CISCO, Common Vulnerability Scoring System Q & A, **http://www.cisco.com/en/US/about/security/intelligence/cvss-qandas.html**

→CVE : **http://www.cve.mitre.org**

→NIST, DHS *add national vulnerability database to mix* : **http://www.securityfocus.com/news/11278**

Organiser la supervision de la sécurité informatique

Parmi les préoccupations des RSSI, intrusions, actes délictueux ou fraudes aux systèmes de traitement sont des sujets qui deviennent centraux. Les pandémies virales ne sont pas pour autant évincées de leur triste podium. Cependant, dans la hiérarchie des risques, les systèmes comptables et financiers, de gestion commerciale ou de production et les systèmes de données « métier » sont prioritaires.

Pour prévenir, détecter et réagir aux incidents de sécurité, les moyens informatiques permettent d'enregistrer et de remonter quantité d'informations à traiter. Si cette supervision est une préoccupation légitime, elle est aussi considérée comme une activité de cybersurveillance. La légalité des pratiques, comme leur coût en efforts et en moyens, imposent d'organiser rationnellement la supervision de la sécurité.

Nous n'abordons pas dans cet article ce qu'il y a lieu de faire après une intrusion (ça n'est pas le sujet ici, ce fut celui du dossier MISC n°14). Nous rappelons juste quelques points pour continuer à travailler, sans pour cela altérer le déroulement des procédures.

Une première partie de cet article aborde des questions clefs et dimensionnantes :

- Quelles traces collecter, sans tomber dans l'excès de surveillance ?
- Avec quels éléments reconnaître la réalisation d'une infraction ?
- Comment continuer à travailler et superviser, sans perturber une enquête judiciaire ?

Ces besoins éclairés, la supervision de la sécurité informatique doit reposer sur des activités définies. Des engagements de niveaux de service, des tableaux de bords, des procédures sont déterminés en correspondance. C'est l'objet de la seconde partie de cet article.

En dernière partie, des prestations spécialisées sont indiquées pour déléguer la télésurveillance, notamment à destination des PME et PMI.

1. De quel droit vous permettez-vous ?

Précisons en préambule que les données suivantes sont basées sur les textes actuellement en vigueur. Les modifications de la loi peuvent rapidement rendre certains points obsolètes.

Ensuite, notez que chaque cas est unique. Les exemples et interprétations données ci-après sont des cas généraux, et nous trouverons toujours des cas précis où ils ne s'appliqueront pas exactement tel que décrit dans cet article. Sous l'angle juridique, les **« traces informatiques »**, leur collecte et leur traitement sont essentiellement régis en France par la loi dite « Informatique et Liberté » du 6 janvier 1978 modifiée le 7 août 2004.

Les infractions à cette loi sont principalement prévues et reprises par les articles 226-16 à 24 du Code Pénal dans la section « Des atteintes aux droits de la personne résultant des fichiers ou des traitements informatiques ».

Le pénaliste n'expliquera pas aux techniciens comment faire. En revanche, il peut expliquer que constitue une preuve ou une pièce à conviction tout élément ou objet produit devant une juridiction répressive et qui a pour objectif d'attester de la matérialité d'une infraction.

La preuve n'a pas à répondre à une forme ou à des critères précis. C'est leur ensemble qui doit permettre à l'enquêteur et aux magistrats de comprendre le déroulement des faits et l'implication de chaque intervenant.

Dès lors, toute « certification » apportée à la trace (par exemple au moyen de scellements des informations enregistrées par les équipements et de certificats X509), pourra utilement la faire passer du statut de simple donnée à celui de fait établi, lequel pourra être qualifié de preuve le cas échéant.

Légalité de la preuve

Suivant une jurisprudence plusieurs fois répétée, « *aucune disposition légale ne permet au juge répressif d'écarter les moyens de preuve produits par les parties au seul motif qu'ils auraient été obtenus de façon illicite ou déloyale, qu'il leur appartient seulement (...) d'en apprécier la valeur probante après les avoir soumis à la discussion contradictoire.* »

Dés lors, même un système de cybersurveillance entaché d'illégalité peut fournir des éléments susceptibles d'être utilisés par les enquêteurs et les juges. Cela ne veut pas dire que tout est permis, simplement que chaque cas est unique et qu'il appartient à la Justice de se prononcer sur chacun.

Ce que dit la Loi

L'article 1er de la loi tient à préciser que « *L'informatique doit être au service de chaque citoyen. (...) Elle ne doit porter atteinte ni à l'identité humaine, ni aux droits de l'homme, ni à la vie privée, ni aux libertés individuelles ou publiques.* »

Et fort logiquement, l'article 226-16 du Code Pénal commence par « *Le fait, y compris par négligence, de procéder ou de faire procéder à des traitements de données à caractère personnel sans qu'aient été respectées les formalités préalables à leur mise en œuvre prévues par la loi est puni de cinq ans d'emprisonnement et de 300.000 euros d'amende.* »

Yves Le-Hir
Capitaine de Police – SRPJ Toulouse – Enquêtes Financières
Yannick Fourastier
Ingénieur de recherche – EADS CCR – **yannick.fourastier@eads.net**

Ainsi, jouer avec des fichiers peut représenter un gros risque pour l'entreprise, son dirigeant et/ou ses responsables informatiques. Les personnes disposant de droits d'accès privilégiés aux systèmes en vue de leur administration, sont particulièrement concernées.

La loi précise (art. 5) qu'elle s'applique aux traitements de données à caractère personnel [1] dont le responsable est, soit établi en France, soit recourt à des moyens de traitement situés sur le territoire français (hors le simple transit).

Les sous-traitants sont prévus (art. 35). Les mêmes obligations vont s'imposer à eux et au commanditaire ; lequel ne peut se décharger des obligations prévues par la loi.

Des données distinctement qualifiées

Dans le flot des données qui circulent sur un réseau informatique, comment différencier une donnée à caractère personnel, donc visée par la loi d'une donnée qui n'a pas de caractère personnel ?

L'article 2 de la loi du 6 janvier 1978 dit : « *Constitue une donnée à caractère personnel toute information relative à une personne physique identifiée ou qui peut être identifiée, directement ou indirectement, par référence à un numéro d'identification ou à un ou plusieurs éléments qui lui sont propres. Pour déterminer si une personne est identifiable, il convient de considérer l'ensemble des moyens en vue de permettre son identification dont dispose ou auxquels peut avoir accès le responsable du traitement ou toute autre personne.* »

Dans ce cadre, tous les identifiants de machines peuvent être des données à caractère personnel. Nous prendrons l'exemple du téléphone : savoir qu'un numéro s'est connecté à un autre ne permet pas de déterminer quelles personnes physiques ont conversé, mais un numéro de téléphone est clairement rangé dans la catégorie « donnée à caractère personnel ».

Il en est de même pour l'identifiant d'une machine (adresse IP ou autre *nommage* identificateur, y compris les adresses MAC). Il n'indique pas nécessairement qui se servait de l'ordinateur, mais recoupé ultérieurement avec les mesures de protection en place, avec les fichiers de la pointeuse, du parking, etc., il permettra de s'en rapprocher et sera donc considéré comme une « donnée à caractère personnel ».

Ainsi, très vite, nous constatons que les historiques qui vont être conservés, ou les données qui vont être agrégées pour détecter les incidents, vont contenir des « données à caractère personnel ».

Comme la Loi du 6 janvier 1978 dit que :

> « *Constitue un traitement de données à caractère personnel toute opération ou tout ensemble d'opérations portant sur de telles données, quel que soit le procédé utilisé, et notamment la collecte, l'enregistrement, l'organisation, la conservation, l'adaptation ou la modification, l'extraction, la consultation, l'utilisation, la communication par transmission, diffusion ou toute autre forme de mise à disposition, le rapprochement ou l'interconnexion, ainsi que le verrouillage, l'effacement ou la destruction.* » (art. 2) ; et

> « *Constitue un fichier de données à caractère personnel tout ensemble structuré et stable de données à caractère personnel accessibles selon des critères déterminés.* » (art. 2) ; nous voila donc dans l'obligation légale de respecter les formalités préalables à la mise en œuvre de ces traitements.

Les seuls cas où les fichiers ou traitements contenant des données à caractère personnel n'ont pas à être déclarés sont limitatifs. Les détails sont sur le site de la CNIL (**www.cnil.fr**), mais nous pouvons résumer en disant que, de façon générale, seuls les traitements comptables sont dispensés de déclaration préalable, ainsi que les données privées de particuliers et les listes des membres d'organisations religieuses, politiques, syndicales ou philosophiques.

De même, certaines prises de positions de la CNIL peuvent s'interpréter comme « surveiller le réseau, oui ; surveiller les personnes, non ».

Les systèmes de surveillance statistiques, automatiques, à anonymisation rapide (« fichiers de journalisation ») non adossés à des traitements de données à caractère personnel et n'ayant pas pour finalité la surveillance des salariés, sont autant de fichiers dont la déclaration n'est pas nécessaire, et ce même s'ils collectent des données jusqu'à l'équipement du salarié.

Une « fiche de synthèse » de la CNIL datée du 11 février 2002 est titrée « Cybersurveillance sur les lieux de travail ». Les responsables d'entreprise, les responsables informatiques et les administrateurs, comme les représentants du personnel pourront utilement s'y reporter tant elle recoupe le présent dossier.

Nous en retiendrons deux prises de positions de la CNIL :

> « Les administrateurs qui doivent veiller à assurer le fonctionnement normal et la sécurité des réseaux et systèmes sont conduits par leurs fonctions mêmes à avoir accès à l'ensemble des informations relatives aux utilisateurs (messagerie, connexion à internet, fichiers 'logs' ou de journalisation, etc.) y compris celles qui sont enregistrées sur le disque dur du poste de travail. Un tel accès n'est contraire à aucune disposition de la loi du 6 janvier 1978. »

> « Aucune exploitation à des fins autres que celles liées au bon fonctionnement et à la sécurité des applications des informations dont les administrateurs de réseaux et systèmes peuvent avoir connaissance dans l'exercice de leur fonction ne saurait être opérée, d'initiative ou sur ordre hiérarchique. »

1 La surveillance conduit à constater une infraction : L'infraction étant soit soupçonnée soit établie, que faire ?... Sauvegarder !

Il n'y a rien de plus volatil qu'une trace informatique. Sauvegardez et archivez les éléments qui vous ont conduit à constater des faits. On ne reprochera jamais de prendre des mesures conservatoires en vue d'une action en justice !...

Lorsque les responsables de l'entreprise intéressés par la commission de l'infraction (responsable informatique pour les constatations, DRH pour les implications avec le Code du Travail, dirigeant qui prendra la décision finale...) auront arrêté une position commune, et rapidement si possible, il conviendra de s'y tenir.

Si c'est une attaque ou une intrusion subie, qui vous cause préjudice et vous est dommageable, vous êtes une victime. Dès lors, vous déposez plainte si vous souhaitez que le ou les auteurs de l'infraction soient identifiés et punis par la justice pénale.

Si l'infraction ne vous cause pas préjudice (par exemple sans effet sur votre réseau ni votre sécurité), pouvez-vous révéler ces faits à la justice ? La loi n'impose aucune obligation générale et absolue au citoyen pour révéler toute infraction dont il aurait connaissance. Hormis la non-dénonciation de crime ou d'atteinte à l'intégrité corporelle, ou visant les mineurs et personnes vulnérables, c'est une possibilité qui est offerte à chacun, un choix moral.

Un point important : la loi n'impose pas à chacun de savoir distinguer une contravention, un délit ou un crime. Vous n'avez pas à déterminer avec exactitude l'infraction que vous pensez exister, mais seulement à faire preuve de bonne foi dans votre révélation (ou témoignage). Si par la suite, les faits n'apparaissent pas constitutifs d'une infraction pénale, vous ne pouvez être pénalement tenu pour responsable des conséquences de votre révélation si vous avez toujours été de bonne foi (Remarque technicienne : les activités de cybersurveillance servent à cela).

Ceci posé, le responsable du traitement informatique (normalement le dirigeant, le DSI, le responsable informatique, l'informaticien, l'administrateur ou selon les délégations de responsabilités le RSSI mais aussi, tout simplement et souvent, surtout les utilisateurs bureautiques) sait si son fichier de cybersurveillance va contenir ou non des données à caractère personnel.

S'il faut le déclarer, rendez-vous sur le site de la CNIL. Nous ne paraphraserons pas inutilement les pages de la CNIL ici. Autant vous reporter à l'original.

S'il faut résumer, disons qu'il va falloir passer par les points suivants :

- Définir la finalité du fichier : savoir quelles données sont collectées et dans quel but, car l'article 226-21 du Code Pénal dit que tout détournement de finalité est puni.

Si je collecte à des fins statistiques les e-mails de tous ceux qui remplissent un formulaire sur mon site, je ne peux pas réutiliser cette liste pour envoyer des messages commerciaux.

- Être transparent : l'article 226-18 du Code Pénal dit que la collecte de données à caractère personnel par un moyen frauduleux, déloyal ou illicite est puni. Et l'article 32 de la Loi énumère les obligations aux responsables de traitement. Au minimum, qui collecte des données, et pourquoi (responsable du traitement ou représentant, et finalité de la collecte)?

Par exemple, que BRZGH Telecom collecte les données informatiques issues de la connexion aux fins d'assurer un meilleur service à ses clients ; ou que JKLM Finance collecte les informations nominatives des visiteurs de son site dans le but de leur proposer la plus haute sécurité possible. Car, dans la ligne de ce texte, le décret 81-1142 du 23.12.1981 précise que **le refus ou l'entrave au bon exercice des droits** des personnes sont également puni, de contraventions de cinquième classe.

- Sécuriser les données : tout fichier de données à caractère personnel doit être sécurisé. L'article 226-17 du Code Pénal prévoit la responsabilité du propriétaire du fichier s'il est, par exemple, pillé et réutilisé frauduleusement et que l'on démontre ensuite que c'est la faiblesse de la sécurisation qui a permis l'intrusion. Si je collecte les numéros de carte bancaire de mes clients, je ne laisse pas la sauvegarde du fichier sur un coin de mon bureau.

- Ne pas divulguer les données à n'importe qui : l'article 226-22 du Code Pénal dit que la communication d'informations à des personnes non autorisées, y compris par négligence ou imprudence, est puni. C'est cohérent avec l'article précédent.

Si j'ai un problème avec le traitement des requêtes, je ne vais pas transmettre le contenu du fichier au « copain du copain » qui va débuguer le process.

- Ne pas conserver les données ad vitam aeternam : la définition initiale du fichier doit inclure une durée prévisible de traitement et de conservation. L'article 226-20 du Code Pénal **sanctionne** la conservation des données pour une durée supérieure à celle qui a été déclarée.

Corollaire du « droit à l'oubli » maintes fois rappelé par la CNIL, la durée de conservation des données à caractère personnel est toujours examinée par la Commission. La Loi Informatique et Liberté ne fixe pas de durée précise. Elle dit simplement que les données ne sont conservées que « *pendant une durée qui n'excède pas la durée nécessaire aux finalités pour lesquelles elles sont collectées et traitées* ».

Par exemple, pour des données liées à la scolarité, la CNIL a accepté la durée demandée de deux ans, ce délai permettant de gérer l'année scolaire en cours et de préparer la suivante. On en revient à la finalité du fichier. De celle ci doit découler une durée suffisante pour le traitement mais non excessive par rapport à celui ci.

Parmi les points à préciser immédiatement sur la cybersurveillance : **le courrier électronique**.

La loi est claire mais souvent outrepassée par les responsables d'entreprises ou leurs **subordonnés** : « *Le fait, commis de mauvaise foi, d'ouvrir, de supprimer, de retarder ou de détourner des correspondances arrivées ou non à destination et adressées à des tiers, ou d'en prendre frauduleusement connaissance, est puni d'un an d'emprisonnement et de 45.000 euros d'amende. Est puni des mêmes*

Des entreprises peuvent-elles être dans l'obligation de stocker des données?

La LCEN (Loi pour la Confiance dans l'Économie Numérique) impose aux Fournisseurs d'Accès Internet de conserver les données de connexion de leurs clients aux fins de répondre aux requêtes des autorités judiciaires. Problème, la durée de cette conservation, comme la nature exacte des données, doivent être fixées par un décret d'application qui n'a toujours pas été publié. La seule chose aujourd'hui fixée est la durée maximale de conservation, soit un an.

Cependant, un arrêt du 4 février 2005 de la Cour d'Appel de Paris a jugé qu'une entreprise offrant à ses salariés un accès internet devait être soumise aux mêmes règles que les FAI en matière de conservation des données de connexion de leurs salariés. Si cette décision est confirmée, elle aura des répercussions majeures sur toutes les entités offrant des accès internet (les cybercafés entre autres).

peines le fait, commis de mauvaise foi, d'intercepter, de détourner, d'utiliser ou de divulguer des correspondances émises, transmises ou reçues par la voie des télécommunications ou de procéder à l'installation d'appareils conçus pour réaliser de telles interceptions. » (Article 226-15 Code Pénal). Donc, toute personne qui consulte le mail d'une autre sans son autorisation ou sans l'excuse de l'étourderie, de l'erreur de destinataire, etc. commet un **délit prévu** et **réprimé**. Toute action de cybersurveillance portant sur le courrier devra donc être clairement définie puis portée à la connaissance des personnes concernées.

Que dire de **l'atteinte à l'intimité de la vie privée** ?

L'article 226-1 du Code Pénal dit : « *Est* **puni d'un an d'emprisonnement et de 45 000 euros d'amende** *le fait, au moyen d'un procédé quelconque, volontairement de porter atteinte à l'intimité de la vie privée d'autrui :*

1 *En captant, enregistrant ou transmettant, sans le consentement de leur auteur, des paroles prononcées à titre privé ou confidentiel ;*

2 *En fixant, enregistrant ou transmettant, sans le consentement de celle-ci, l'image d'une personne se trouvant dans un lieu privé.*

Lorsque les actes mentionnés au présent article ont été accomplis au vu et au su des intéressés sans qu'ils s'y soient opposés, alors qu'ils étaient en mesure de le faire, le consentement de ceux-ci est présumé. »

Il est suivi de l'article 226-2 qui précise : « *Est puni des mêmes peines le fait de conserver, porter ou laisser porter à la connaissance du public ou d'un tiers ou d'utiliser de quelque manière que ce soit tout enregistrement ou document obtenu à l'aide de l'un des actes prévus par l'article 226-1.* »

Et l'on retrouve les fichiers de données à caractère personnel dans l'article 226-22 qui dit que « *Le fait, par toute personne qui a recueilli, à l'occasion de leur enregistrement, de leur classement, de leur transmission ou d'une autre forme de traitement, des informations nominatives dont la divulgation aurait pour effet de porter atteinte à la considération de l'intéressé ou à l'intimité de sa vie privée, de porter, sans autorisation de l'intéressé, ces informations à la connaissance d'un tiers qui n'a pas qualité pour les recevoir est puni d'un an d'emprisonnement et de 15 000 euros d'amende.*

La divulgation prévue à l'alinéa précédent est punie de 7 500 euros d'amende lorsqu'elle a été commise par imprudence ou négligence ».

Si nous croisons ce point et les précédents, nous constatons que mettre en place un suivi des mails entrant et sortant, pour s'assurer que les salariés ne passent pas leur temps à envoyer et recevoir des mails et des pièces jointes, en particulier hors de l'entreprise, va devoir donner lieu à déclaration à la CNIL si les adresses mail ou identifiants des machines sont conservées ; et pas forcément si on se contente de collecter les domaines ou fournisseurs d'accès et les volumes échangés, mais qu'il est de toute façon formellement interdit d'accéder au contenu des courriers. On pourra objecter que les logiciels antivirus ou anti-spam le font et peuvent, suivant leur paramétrage, mettre en quarantaine ou isoler des courriers jugés dangereux ou indésirables, que des personnes physiques différentes du destinataire pourraient alors consulter. Si une action en justice est introduite un jour sur une telle base, le résultat en sera intéressant.

Un dernier point sur la Loi Informatique et Liberté, à propos des **« Correspondants Informatique et Liberté »** (CIL).

Les modifications du 7 août 2004 ont prévu ce nouvel intervenant. En contrepartie de la désignation d'un correspondant informatique et liberté, il y a allègement des obligations déclaratives. En gros, c'est de l'autocontrôle : le correspondant informatique et liberté devient le garant du respect des obligations relatives aux fichiers de données à caractère personnel. Cependant, à ce jour, les décrets d'applications ne sont toujours pas parus (mais ils ne sauraient tarder), donc nous sommes toujours dans un cadre sans correspondant informatique et liberté.

Et **en entreprise** ?

La vie privée est également protégée au sein de l'entreprise. Le Code du Travail par ses articles 120-2, 121-8 et 432-2, dit successivement que :

- *Nul ne peut apporter aux droits des personnes et aux libertés collectives des restrictions qui ne seraient pas proportionnées au but recherché ;*
- *Aucune information concernant personnellement un salarié ou un candidat à l'emploi ne peut être collectée par un dispositif qui n'a pas été porté préalablement à la connaissance du salarié ou du candidat à l'emploi ;*
- *Le comité d'entreprise (...) est informé et consulté, préalablement à la décision de mise en œuvre dans l'entreprise, sur les moyens ou les techniques permettant un contrôle de l'activité des salariés.*

On constate que ces dispositions reprennent l'esprit de protection de la vie privée ci-dessus évoqué, et les obligations portant sur les traitements de fichiers de données à caractère personnel.

Reconnaître la réalisation d'une infraction

De façon réaliste et pratique, il n'y a guère que la définition d'une activité « moyenne » d'un salarié ou d'un poste de travail qui pourra nous orienter, s'il s'agit de mettre en place un premier niveau automatique de tri. Certes, l'informaticien peut toujours

tomber « par hasard » sur une infraction : une intervention sur un poste, un mail qui s'égare, etc. ; mais ce ne sont pas des recherches automatisables. En fonction de son réseau et de son activité, chaque administrateur choisit ses critères pertinents et ce qui s'écarte trop de la fourchette donne lieu à contrôle. Ce sont des critères de rapport coût/résultat qui vont entrer en compte.

Les infractions les plus fréquemment constatées :

➜ **intrusions :**

votre réseau est utilisé comme point de départ pour des attaques, dénis de service et autres. Les DOS génèrent souvent des traces caractéristiques, répétitives. Quant aux attaques, c'est le choix du protocole ou des ports qui permettra de les soupçonner.

➜ **serveur de fichiers contrefaisant :**

votre réseau héberge à votre insu des zones d'hébergement (diffusion ou rapatriement) de fichiers images, musicaux, vidéo ou autres. Là, c'est le nombre de connexions extérieures et le volume des échanges qui sera significatif. Les autres infractions réalisables à travers un réseau ne génèrent souvent que des traces trop faibles pour être aisément discriminées de l'activité habituelle. C'est typiquement le cas des fraudes sur les systèmes de gestion. Leur détection implique que les applications correspondantes aient prévu des étapes de sécurité avec des points de contrôle des traitements et de croisements loguant.

De nombreuses autres utilisations du réseau peuvent contrevenir aux règles de bon usage de l'entreprise mais, sans infraction visée par le Code Pénal, elles n'entrent pas dans le présent cadre.

Les besoins légitimes et par obligation ainsi récapitulés constituent une base importante pour définir les traces à constituer.

Les informations à collecter techniquement et à agréger, comme les actions de protection et de conservation, peuvent être définies.

Aussi, nous pouvons identifier des activités pour :

1. **distinguer les « traces sécurité »** des autres correspondantes aux conditions de bon fonctionnement ;
2. **spécifier les informations** d'application comme les données techniques pertinentes pour chacun des besoins de trace ;
3. **arbitrer leur traitement** selon les contraintes juridiques les concernant, individuellement et agglomérées à d'autres ;
4. **décider des moyens** de collecte, de traitement, de conservation et de destruction adaptés, tout comme des réactions.

2. Organiser... et s'organiser en pratique

Dans l'alignement de ces activités, la supervision de la sécurité informatique doit s'organiser pour trois raisons :

1. Le coût induit par les systèmes de supervision (matériels et logiciels) comme par les efforts humains ne sont pas à négliger.
2. Pour être utile, la supervision implique d'être pertinente.
3. Les risques juridiques de la cybersurveillance l'amène à devoir être cohérente avec les contraintes et les obligations rappelées ci-dessus.

Organiser la maîtrise des coûts

Manager la sécurité, corréler les logs, etc. sont autant de thèmes marketing réchauffés au fil des saisons et au gré du vent soufflant dans certaines réunions « technico-commerciales ». Superviser la sécurité est une activité coûteuse, mais utile en rapport des risques considérés. Sa « rentabilité » (rapport Coût/Risque) peut être considérée en fonction de son efficience, soit le rapport coût/efficacité.

Aussi, pour simplifier, nous considérerons ici seulement les coûts relatifs aux aspects :

➜ **techniques :**

matériels, logiciels, maintenances ;

➜ **organisationnels :**

soit l'équivalent jours/hommes par compétence.

Nous ne paraphraserons pas ici l'article « Gestion des risques et maîtrise des coûts » [MISC 7], préférant un bref rappel.

Maîtriser les coûts relatifs à la « technique » est un sujet généralement bien traité.

L'utilisation de solutions libres (par exemple telles celles présentées dans la suite de ce dossier) pour réaliser les fonctions de la supervision, outre défendre une position déontologique franche, contribue à cette maîtrise de façon certaine. La contractualisation de la maintenance avec les sociétés de services suffisamment courageuses et honnêtes pour réaliser le relais nécessaire avec la communauté (via l'emploi de vraies compétences correctement payées) assure une alternative pleinement équivalente aux solutions propriétaires croulantes sous les brevets généralement abscons et dangereux.Les coûts relatifs à l'organisation humaine sont en revanche souvent à maîtrise plus dérivante. Si les choix techniques ont leur part de responsabilité, notamment du fait de la maturité technologique et du degré d'intégration fourni, la façon de réaliser les tâches par les bonnes compétences (et de bonne volonté) est une équation plus délicate à résoudre. L'organisation des activités autour de la supervision, avec des tâches définies et allouées en fonction des compétences, permet d'adresser correctement ce problème.

Pertinence de la supervision

Une bonne supervision doit commencer par fournir les bonnes informations pour la gestion des traces attendues. Nous entrons là dans un projet de système d'information tout ce qu'il y a de plus habituel... Il est juste spécifique à un périmètre fonctionnel particulier.

Par cohérence avec le dossier, la suite de cet article traite les traces de malveillances sur des réseaux informatiques.

Cette supervision consiste à détecter le moment de réalisation d'une infraction. Soit du fait de comportement anormal du

trafic, soit des systèmes (notamment dans le cas d'une injection « cohérente » via les couches applicatives pour générer des fautes). Aussi, les données intéressantes peuvent provenir du réseau via les IDS, les alertes de filtrage (ACL de routeurs, FW, proxies, *socks*), les sondes SNMP des commutateurs (très utiles pour détecter des croissances de trafic anormales, signe possible d'extrusion d'information), ou des logs divers... et toujours très variés !

Aussi, des points de mesure doivent être définis pour implanter les « capteurs ». Selon les chemins possibles, la corrélation d'alerte et le niveau de risque envisagé peuvent varier, influant du coup la supervision. L'étude fine et précise de ces points et de ce qu'ils doivent exactement remonter, permet déjà de disposer d'une supervision saine. La pertinence va provenir principalement de la corrélation, c'est-à-dire la mise en relation des évènements collectés. En définissant rigoureusement ces règles de couplage, outres des relations binaires (couples), l'étude de caractéristiques pour identifier les équivalences possibles permet de réduire le taux de faux-positifs et de vrais-négatifs. Dans la pratique, à moins d'être un ayatollah de la théorie des graphes, on se contente d'indiquer au système d'apprentissage (lorsqu'il y en a un...) ce qu'il peut considérer comme des fausses alertes.

S'organiser en tenant compte des risques juridiques et autres

Collecter des traces malveillantes, c'est légitime... mais ça doit rester légal. Exclure de l'ensemble des traces collectées, celles non légales n'est pas trivial.

Ceci principalement pour des raisons techniques d'identification de ces dernières, et de retrait sans impact en intégrité, complétude et cohérence des autres traces. Par ailleurs, selon le type de trace, les conditions de traitement, de stockage et de destruction peuvent nécessiter des précautions particulières : stockage en accès restreint, chiffrement, effacement sécurisé, etc. Faites attention aux temps de conservation des traces : préférez l'utilisation de médias de type CD-Rom non réinscriptible. De plus, certaines technologies à obsolescence rapide ou en fin de cycle pourraient être handicapante (par exemple les disques de type DON). Concernant le scellement des traces pour des vérifications d'intégrité : utile au responsable (par défaut, le dirigeant, mais bien souvent l'administrateur) pour montrer sa volonté de mettre en œuvre tous les moyens permettant de montrer sa bonne foi, le pénaliste lui ne considère pas sa fiabilité. En effet, vous fournissez les traces et les clefs correspondantes... que **vous** avez générées ! Du moins, **vos** équipements.

De là à externaliser votre supervision à un **tiers de confiance**...

Enfin, la supervision implique une acquisition et un traitement d'information particulièrement importants et généralement très obscurs (une fois installées, plus personne ne se souciera des boîtes noires). Pour l'intégration sur votre réseau, comme pour l'exploitation de votre supervision ou son externalisation, **ne la confiez pas à n'importe qui**.

Les activités de supervision, en particulier de la sécurité

Pour aller à l'essentiel, les activités de supervision informatique sont standardisées. Vous trouverez une littérature abondante sur le sujet avec la boîte à outil ITIL (*IT Infrastructure Library*). Cet ensemble documentaire regroupe des procédures génériques, des plans documentaires standards et des conseils pertinents pour améliorer l'efficacité des systèmes d'information, optimiser les activités de production informatique (supervision, administration, *help desk*, achats, etc.), améliorer la qualité des services

2 Détection d'une infraction à caractère public : l'infraction est publique : pas de difficulté.

C'est le cas par exemple d'une petite annonce accessible à tous sur l'intranet de l'entreprise et qui proposerait les derniers logiciels à des prix défiant toute concurrence. Vous avez le droit (mais non l'obligation) de révéler. C'est un choix moral.

Craignez-vous d'être impliqué comme complice si vous ne révélez pas ?

La complicité est définie par l'article 121-7 du Code Pénal, qui dit que « *Est complice d'un crime ou d'un délit la personne qui sciemment, par aide ou assistance, en a facilité la préparation ou la consommation. (...)* ».Il est à supposer que vous ne vous placerez jamais dans la position où votre action (ou absence d'action) permettra la consommation ou la préparation de l'infraction. On peut placer ici la position du gestionnaire de forum : à partir de quel moment peut-il craindre de voir sa responsabilité mise en cause ? Dés lors qu'il aura sciemment fourni aide ou assistance. En pratique, c'est toujours une décision découlant d'un jugement personnel de la nature de la situation.

Est-ce que l'absence d'effacement d'un message à caractère frauduleux est une assistance à la préparation ou à la consommation d'une infraction ?

La jurisprudence nous dit qu'une simple négligence ne peut être assimilée à une participation intentionnelle ; que l'élément intentionnel implique que son auteur ait eu conscience de l'aide apportée à l'action principale ; il implique une participation volontaire et consciente de l'aide apportée à la commission d'une infraction.

Maintenant, il existe une qualification pénale qui s'appelle le recel (article 321-1 du Code Pénal) qui se définit par « (...) *le fait de dissimuler, détenir ou de transmettre une chose, ou de faire office d'intermédiaire afin de la transmettre, en sachant que cette chose provient d'un crime ou d'un délit. (...) également (...) le fait, en connaissance de cause, de bénéficier, par tout moyen, du produit d'un crime ou d'un délit.* »

Détenir, utiliser ou tirer profit de l'utilisation, en toute connaissance, par exemple de programme contrefaisant, fait de vous un receleur. Notons que dans le cas d'une infraction visant l'intégrité corporelle d'une personne, outre le cas des mineurs et personnes vulnérables déjà citées, la non-assistance à personne en danger est prévue et réprimée par l'article 223-6 du Code Pénal. Dans ce cas, il y a obligation de révéler les informations dont vous avez connaissance.

Ce serait par exemple le cas d'un mail dont vous prendriez connaissance par erreur et qui contiendrait des éléments suffisamment clairs et plausibles pour vous faire penser que si c'était vous la personne visée vous aimeriez bien être protégé.Encore plus, l'article 434-1 du Code Pénal punit celui qui n'informe pas les autorités judiciaires « *d'un crime dont il est encore possible de prévenir ou limiter les effets ou dont les auteurs sont susceptibles de commettre de nouveaux crimes qui pourraient être empêchés* ».

Révéler un crime alors qu'il peut encore être évité est donc une obligation.

3 Détection d'une infraction à caractère restreint « au privé » ou confidentiel : L'infraction n'est pas publique : avez-vous le droit de la révéler ?

Le « secret professionnel » est défini par l'article 226-13 du Code Pénal : « *La révélation d'une information à caractère secret par une personne qui en est dépositaire soit par état ou par profession, soit en raison d'une fonction ou d'une mission temporaire, est punie d'un an d'emprisonnement et de 15000 euros d'amende.* »

Donc, si vous avez connaissance d'une infraction alors que vous agissez dans ce cadre, vous voila lié au secret, vous n'avez pas le choix. Notons que la loi ne prévoit pas expressément les personnes, professions ou fonctions. La jurisprudence pénale est pléthorique pour les médecins, avocats, ministres des cultes, experts-comptables, notaires ou autres, mais il semble qu'au pénal il ne soit nulle part fait mention d'informaticien, administrateur ou toute autre profession liée à l'informatique.

Ce que vise la loi, c'est la confiance qui s'impose dans certaines professions, notamment la sécurité des confidences qu'un particulier est dans la nécessité de faire à une personne dont l'état ou la profession fait d'elle un confident nécessaire.

Dans le cas général (par exemple une intervention sur un disque dur en panne), c'est une obligation de discrétion, liée à la notion de protection de la vie privée, qui prévaut. Dès lors, autant la partie dénoncée peut relever un manquement qu'elle poursuivra au Civil, autant la partie dénonciatrice ne semble pas pouvoir être poursuivie au plan pénal. Révéler une infraction non publique est un choix moral.

Concernant le secret professionnel, vous en êtes délié par l'article 226-14 du Code Pénal qui prévoit que « *L'article 226-13 n'est pas applicable dans les cas où la loi impose ou autorise la révélation du secret. En outre, il n'est pas applicable (...) 1°: A celui qui informe les autorités judiciaires, médicales ou administratives de privations ou de sévices, y compris lorsqu'il s'agit d'atteintes sexuelles, dont il a eu connaissance et qui ont été infligées à un mineur ou à une personne qui n'est pas en mesure de se protéger en raison de son âge ou de son incapacité physique ou psychique* ».

Vous en êtes également délié par l'article 434-1 du Code Pénal réprimant la non-dénonciation de crime. Dans les deux cas, il s'agit d'une possibilité de révélation, non d'une obligation.

La loi pénale a donc prévu des circonstances très précises quant au secret ou à la révélation du secret. Pour les autres cas, infraction publique ou pas, c'est une responsabilité morale.

informatiques. Comme beaucoup de normes et de standards, cette boîte à outils est malheureusement payante. La supervision est traitée via les aspects d'*incident/problem management*. La supervision de la sécurité, toute particulière qu'elle soit, n'échappe pas à cette règle. Elle nécessite de plus grandes précautions, en confidentialité notamment.

Pour résumer ces activités **normalisées** et en simplifiant, elles adressent :

➜ **la signalisation :**

la détection d'une alarme ;

➜ **le diagnostic et la qualification :**

la vérification de l'alarme permettant d'indiquer la réalité de l'incident ;

➜ **l'assignation et le traitement :**

l'application éventuelle de parades définies pour une première réaction en urgence ;

➜ **la clôture :**

la compréhension de l'incident, éventuellement l'enquête, etc.

Définir comment sont réalisées chacune de ces activités implique de préciser le rôle de chacun des intervenants, les outils dont il a besoin et les résultats qu'il doit produire sous conditions de présentation, voire sous contrainte de réactivité ou d'astreinte.

Ce travail d'organisation dépasse la simple description des moyens techniques et de points d'implantation de diverses sondes. Pour être rigoureux, s'inscrivant en ligne notamment avec une législation pénale, il nécessite d'être clair sur les engagements de moyens et/ou de résultats, comme sur les responsabilités de chacun.

Engagements de services

Les besoins de supervision sont comme les besoins de sécurité : ils varient selon les environnements techniques, les chaînes de traitement et surtout les applications. En particulier, concernant les malveillances, les efforts de supervision doivent être cohérents avec la ségrégation des environnements homogènes en besoins de sécurité. Par exemple, des engagements de services organisés sur des heures ouvrables pour une chaîne de traitement donnée, amènent de ne pas « voir » les incidents hors cette plage. Les conséquences sur une chaîne de traitement immédiatement adjacente, de sensibilité et d'engagements supérieurs (par exemple H24), induisent une situation de crise par incohérence... Le degré d'isolation de ces environnements adjacents peut en effet éviter quelques moments de stress inutiles dans la réaction. La présence d'un serveur web pour lequel les engagements de service sont « légers » (et pour lequel la robustesse n'aura probablement pas été renforcée), à côté du serveur de gestion financière, supervisé, mais sans cloisonnement peut générer des situations ubuesques...

Même si les engagements de service de cette application sont élevés, la détection d'incident alertera certainement un peu trop tard. Le curseur du niveau de stress de l'équipe de supervision risque alors de flirter avec le cran « Panique », impactant la qualification de la gravité réelle et le post-traitement.

Gestion par contrats et tableaux de bords

Avec des besoins de supervision identifiés, des engagements correspondants définis, la façon de les réaliser peuvent faire l'objet de contrats de service. Ceux-ci consistent à clarifier les prestations (éléments à superviser et besoins, temps de réaction, traitements de crise prévus, etc.) à exécuter sur un environnement considéré. Ces contrats précisent également des notions de responsabilités, des conditions d'exécution comme de limites de services. Ce type de gestion par contrat permet à une cellule de supervision d'organiser ses pratiques en niveaux homogènes. La configuration de l'offre de service correspond à ce qu'elle sait réaliser de façon « industrielle » avec les moyens dont elle dispose. Pour ne pas nous attarder sur ces aspects, vous trouverez plus d'information sur les *Managed Services* dans l'ensemble documentaire ITIL. Les tableaux de bords remontent

4 La finalité légale sur laquelle la cybersurveillance doit être alignée : Déposer une plainte ?

COMMENT ?

Directement au Commissariat de Police ou à la Brigade de Gendarmerie locale.
C'est rapide et facile, mais peut vous exposer à un refus ou à un mauvais traitement (par manque de compétence ou surcharge de travail). Néanmoins, tout service recevant les plaintes est tenu de recevoir et d'enregistrer toute plainte, quitte à transmettre ensuite à un service plus compétent. Vous vous exposez alors à devoir attendre la saisine d'un autre service, et à devoir tout répéter à un nouvel interlocuteur. Il est à noter que, face à la monté des plaintes liées à la cyber-délinquance les services de Sécurité Publique se dotent peu à peu de « référents cybercriminalité » dotés des connaissances suffisantes pour vous recevoir et apprécier la suite à donner à votre démarche.

Cette démarche est en cours d'installation, ces « référents » peuvent ne pas encore exister partout.

Par lettre-plainte auprès du Procureur de la République
Le Procureur de la République dispose du privilège de l'opportunité des poursuites : c'est lui qui peut classer sans suite ou faire procéder à une enquête, par le service de son choix (Police, Gendarmerie, service judiciaire spécialisé), mais il y a risque de perte de temps (traitement par le Tribunal) et risque de perte de tout contrôle sur le cheminement de la plainte (choix du service enquêteur, des investigations à réaliser...).

Par plainte avec constitution de partie civile.
Par l'intermédiaire d'un avocat, auprès du Doyen des Juges d'Instruction. Cela entraînera automatiquement l'ouverture d'une information judiciaire et la désignation d'un juge, mais prévoyez des délais et des frais d'avocat.

Après contact préalable, auprès d'un service ou d'un enquêteur spécialisé.
Cela vous permet d'avoir un interlocuteur compétent : les E.S.C.I. (Enquêteurs Spécialisés en Criminalité Informatique) dans chaque Service Régional de Police Judiciaire ; les enquêteurs N-Tech dans les Brigades de Recherche en Gendarmerie ; et à Paris les policiers de la B.E.F.T.I. (Brigade d'Enquête sur les Fraudes aux Technologies de l'Information, dépendant de la Police Judiciaire).

LES CADRES JURIDIQUES POSSIBLES

L'enquête de flagrant délit.
Dans les 48 heures après la constatation de l'infraction, dure au maximum 7 jours ; les enquêteurs disposent de pouvoirs coercitifs (sous contrôle, toujours, du procureur). Si vous êtes sous le coup d'une attaque, si l'infraction est toujours en cours, bref pour tous les cas d'urgence.

L'enquête en mode préliminaire.
La plus courante, sans délais contraignants (pour rappel et sauf cas particuliers la prescription d'une contravention est d'un an, celle d'un délit de trois ans et celle d'un crime de dix ans). Les enquêteurs disposent peu de pouvoirs coercitifs. C'est le cadre général des enquêtes ouvertes sur instruction du procureur.

L'enquête en exécution de Commission Rogatoire.
Délivrée aux enquêteurs par le juge d'instruction, après ouverture d'une information judiciaire (sur réquisitoire introductif du Parquet ; après plainte avec constitution de partie civile ou après une enquête « classique »).

OÙ DÉPOSER UNE PLAINTE ?

Les critères de détermination du service d'enquête et du Tribunal compétent sont, dans l'ordre : la localisation de l'auteur, le lieu de l'infraction, la localisation de la victime.

COMMENT BIEN DÉPOSER UNE PLAINTE?

Savoir faire la différence entre acte malveillant et incident, bug, erreur, acte involontaire (cas des virus intrusifs avec dissémination des données).

Sauvegarder les traces : sauvegardez les logs ; ne réutilisez pas le matériel en cause ; isolez-le du reste du réseau et ne l'éteignez pas...

Agir vite : il y a déperdition rapide des traces en informatique et peu de conservation des données chez les différents intervenants (24 H pour les *proxys* internet, 3 mois pour les *logs* d'IP chez beaucoup de FAI, et de toute façon 1 an délai maximum de conservation des fichiers de données à caractère personnel selon la CNIL).

Ne pas contacter l'auteur de l'acte malveillant : ne pas lui dire qu'une plainte est déposée, ne pas l'induire à effacer les traces de ses actes dans ses matériels.

QUE VONT VOUS DEMANDER LES ENQUÊTEURS ?

Et bien, cela dépendra de chaque cas. Des règles de conduite générale et absolue n'existent pas, elles dépendent de chaque infraction, de chaque enquête et de circonstances particulières que seuls les enquêteurs sauront estimer.

À côté de sauvegardes inaltérables (typiquement des gravures CD) contenant si possible tous les éléments informatiques descriptifs des faits, il sera pris par les enquêteurs des auditions détaillées permettant de comprendre ultérieurement ces sauvegardes, de les expliquer à un public qui n'est pas forcément constitué de spécialistes.

TOUT CECI VA-T-IL BLOQUER VOS ÉQUIPEMENTS ET VOS INFORMATICIENS ?

Et bien, encore une fois, cela dépendra de chaque cas. Comprenez bien que pour une recherche de traces dans votre système propriétaire dans une enquête portant sur un enlèvement d'enfant, les contraintes seront plus fortes que dans une enquête de déni de service dont vous êtes la victime et pour laquelle vous avez déjà sauvegardé tous les logs d'attaque. Plus l'affaire sera technique, plus on rentrera dans votre système et plus votre implication sera demandée par les enquêteurs. Mais il est vraisemblable que si les enquêteurs ont besoin d'entrer en profondeur dans votre système, vous souhaiterez les accompagner. Vous êtes le garant du bon fonctionnement du système. Quant à « immobiliser » un système, c'est une exigence qui n'est que rarement nécessaire, et qui décroît à mesure de l'importance de cet équipement pour votre entreprise.

QU'EST-CE QU'ON PEUT ME DEMANDER ?

Et bien, si vous êtes la victime, vous aurez sûrement à cœur de fournir aux enquêteurs toutes les informations qui peuvent leur être nécessaire. Si vous êtes dans la position du tiers ou du sachant, les enquêteurs peuvent vous requérir, c'est-à-dire vous obliger, et ce sont eux qui prennent la responsabilité pénale de l'acte demandé.

Ce sont les articles 60-1, 77-1 et 99-3 du Code de Procédure Pénale qui définissent les réquisitions judiciaires suivant le cadre juridique et qui contiennent tous la même formulation : le juge d'instruction, le procureur de la République, l'Officier de Police Judiciaire « *peut requérir de toute personne, de tout établissement ou organisme privé ou de toute administration publique qui sont susceptibles de détenir des documents intéressant l'enquête, y compris ceux issus d'un système informatique ou d'un traitement de données nominatives, de lui remettre ces documents, sans que puisse lui être opposée, sans motif légitime, l'obligation du secret professionnel (...)* ».

Seuls les avocats, médecins, huissiers, avoués, notaires et journalistes peuvent refuser de répondre... pas les responsables informatiques !

Et l'article 60-1 précise que l'absence de réponse dans les meilleurs délais est punie d'amende.

QUI EST RESPONSABLE POUR DIALOGUER AVEC LES ENQUÊTEURS ?

« *Nul n'est responsable pénalement que de son propre fait* ». C'est ce que dit l'article 121-1 du Code Pénal. Mais parce qu'il incombe au chef d'entreprise une obligation légale de surveiller son préposé et de veiller à l'observation des règlements au sein de son établissement, il en est pénalement responsable.

Néanmoins, l'évolution de notre société et des entreprises a permis de dégager une jurisprudence maintenant bien établie autour du concept de « délégation de pouvoir » et « délégation de responsabilité », qui peuvent être résumées par cette reprise du texte de plusieurs arrêts de 1993 de la Chambre Criminelle de la Cour de Cassation : « *Sauf si la loi en dispose autrement, le chef d'entreprise qui n'a pas personnellement pris part à la réalisation de l'infraction, peut s'exonérer de sa responsabilité pénale s'il apporte la preuve qu'il a délégué ses pouvoirs à une personne pourvue de la compétence, de l'autorité et des moyens nécessaires* ». D'où l'intérêt réciproque pour le chef d'entreprise comme pour ses salariés employés à des postes de responsabilité de définir les missions et les moyens attribués. Ce point est important en particulier pour le cas des DSI, des RSSI et notamment des CIL.

QUI VA DISCUTER AVEC LES ENQUÊTEURS ?

Le responsable légal de l'entité sera le nécessaire interlocuteur initial. Puis, bien souvent, il délèguera à l'homme de l'art, qu'il soit un salarié ou un prestataire externe.

Les enquêteurs ne cherchent pas des fonctions, ils cherchent des compétences.

à fréquence contractualisée des états chiffrés correspondants aux environnements supervisés. Typiquement, ces tableaux de bords peuvent indiquer des quantités de vulnérabilités, d'attaques, des répartitions par systèmes, par typologies, en camembert, en diagramme bâton, etc. Bref, une remontée « d'information » aux commanditaires du contrat pour leur dire le plus souvent : « tout va très bien ». Il ne faut cependant pas dénigrer l'utilité de ces graphiques : ils permettent à la supervision de justifier son existence... réellement utile le jour où il y a un véritable incident !

Les solutions techniques décrites dans la suite de ce dossier font pour la plupart de jolis tableaux de bords, probablement tout prêts pour un contrat de service type.

3. « Télésurveillance » pour les PME

Loin de se compliquer la tâche à essayer de mettre en œuvre des usines à gaz et des documentations trop riches pour un environnement informatique raisonnable (comparé à celui de sociétés multinationales), les PME/PMI ont tout intérêt à aller simplement et directement à l'essentiel pour leur sécurité : la configuration « réfléchie » des logs sur les serveurs, applications, antivirus, routeurs, firewall, etc. Des sociétés de services peuvent assister les PME/PMI pour définir une politique de gestion de traces et la configurer sur les équipements. La consultation régulière et l'analyse des traces est ensuite nécessaire... pour que la sécurité soit tout simplement efficace ! Cependant, cette activité d'analyse de logs est généralement trop lourde pour un administrateur dans une PME/PMI. Avec les années, les grandes sociétés ont organisé leur production informatique en convergence vers les standards BS15000 de *Managed Services*. Prestataires de services internes, ils ont inspiré de nouvelles filières économiques pour des opérateurs de services à valeur ajouté comme les MSSP (*Managed Security Services Provider*). Pour les sociétés ne souhaitant pas développer de compétences propres en sécurité informatique, les MSSP leur fournissent ces services. Outre de grands opérateurs comme France Télécom qui proposent des services adaptés en particulier aux PME/PMI, sans leur faire non plus de publicité, nous pouvons citer d'autres acteurs spécialisés comme VIANetworks, Thales Secure Solution, Securalis ou Ubizen.

L'offre de télésurveillance couvre généralement les équipements « de sécurité » : VPN, pare-feu, antivirus, sondes de détection d'intrusions, logs des serveurs, etc.

Particulièrement pertinents pour les PME/PMI, ces opérateurs spécialisés sont en outre un appui fort pour les responsables informatiques internes. Si la délégation de responsabilité est un sujet toujours très sensible (en cas d'ennui lors de conférences SSI, c'est toujours l'occasion de débats garantis), la réalité des services de proximité pour des partenaires de ces MSSP renforce généralement bien le niveau de sécurité technique des PME/PMI. Généralement de petites sociétés de services « internet » ou des SSLL locales sauront promouvoir l'offre de ces MSSP, relayer un premier niveau de prestation technique et agir en interface à transparence progressive pour les prestations à valeur ajoutée. En plus d'être techniquement mûre, la supervision de la sécurité informatique est une réalité possible pour les PME/PMI, notamment sur le plan économique avec un coût de prestation de service très abordable. Pour finir, les Conseils habituels des dirigeants de PME/PMI (avocats, notaires et experts comptables) sont les professionnels historiques de la protection de l'information et du patrimoine. En saine neutralité (le Conseil de l'Ordre veille au respect du code de déontologie), ils indiquent toujours aux responsables légitimes ce qu'ils doivent faire comme ce qu'ils peuvent faire. Cette réalité, souvent oubliée hors des cercles dirigeants, concerne en particulier les aspects juridiques des moyens informatiques. Les consultants technologiques des prestataires de services trouveront là des interlocuteurs incontournables sur les questions de cybersurveillance en entreprise, d'autant que les Ordres concernés sont sensibilisés et attentifs à certaines évolutions.

En guise de conclusion

La cybersurveillance en entreprise est juridiquement encadrée. Tout n'est pas possible pour prévenir les malveillances envers des machines et par là, les affaires économiques au sens large. La protection des individus prime, en particulier celle des libertés individuelles. Les recours judiciaires indiqués informent de manière à ce que les victimes puissent exprimer l'outrage subi et laisser agir en cohérence les représentants de la force publique (Police, Gendarmerie, services judiciaires spécialisés, etc.).

La cybersurveillance s'organise alignée sur ces contraintes et dans cette optique. Autrement, inutile et coûteuse, elle est déviante et peut faire l'objet de sanctions, notamment pénales. Pour agencer les moyens techniques et les pratiques humaines associées, les standards ITIL d'*incident & problem management* peuvent servir de référence utile. Les activités industrialisées amènent une gestion en niveaux de services avec des contrats correctement formatés. Ceux ci clarifient les engagements, les modalités d'intervention et leurs limites, comme ils fixent les conditions de remontées périodiques d'information structurées en tableaux de bords.

Les PME/PMI sont en particulier adressées par le biais d'offres adaptées. Acteurs spécialisés en « télésurveillance de l'informatique », les MSSP peuvent être relayés par les partenaires de proximité usuels des PME/PMI. Enfin, les Conseils habituels pour la protection du patrimoine, régulés par leurs Ordres légitimes (Avocats, Notaires, Experts Comptables), sont les acteurs incontournables pour les PME/PMI. Évolutions technologiques incitant, peut-être leur rôle aurait-il besoin d'être mieux connus pour rapprocher la sécurité des informations, à laquelle ils se consacrent, d'autres domaines de la SSI dont la sécurité informatique ?

[1] Les termes « donnée nominative » de la Loi Informatique et Liberté version 1978 ont été officiellement remplacés par les termes « donnée à caractère personnel », traduction de l'expression anglophone « *personnal data* » utilisée au niveau européen.

Collecte d'informations

Renaud Bidou – **renaudb@radware.com**
Consultant Sécurité Europe/Radware, **http://www.radware.com**
Renaud Deraison – **renaud@rstack.org**

La mise en place d'un SOC nécessite naturellement la collecte et le formatage d'informations. C'est à partir de ces informations qu'il sera possible aux opérateurs de prendre les décisions adéquates. La pertinence et la précision des données remontées, normalement après traitement, sont par conséquent des éléments clefs du bon fonctionnement du centre d'opérations.

Caractéristiques du mécanisme de collecte

Avant de foncer tête baissée dans les problématiques liées à la nature des événements collectés, il est indispensable de prendre en compte deux éléments indispensables intervenant en amont de la collecte :

1. Le mode de collecte ;
2. Le mécanisme de transport.

Pour chacun des types de messages, potentiellement pertinents au titre des opérations du SOC, il est nécessaire de définir ces deux caractéristiques. A défaut de prendre en compte l'ensemble des points évoqués ci-dessus, la collecte s'avérera au mieux anarchique et au pire totalement inefficace.

Mode de collecte des informations

Le mode de collecte est essentiellement une caractéristique temporelle et technique. Elle permet de définir si la collecte doit (ou peut) s'effectuer de manière synchrone ou asynchrone. Il s'agit là d'un élément crucial dans la mesure où certaines informations doivent être traitées rapidement (les alertes provenant d'un IDS par exemple) alors que d'autres nécessiteront une action moins rapide, par exemple les rapports réguliers générés par les outils d'analyse de vulnérabilité.

Bien entendu, il est indispensable de prendre en compte le rôle du SOC. S'il s'agit de réagir en moins de cinq minutes (délais garantis par contrat) le mode synchrone sera privilégié alors qu'un SOC dont le rôle est de maintenir un niveau de sécurité constant du système d'information pourra se contenter d'un mode asynchrone.

Mécanisme de transport

Choisir (ou subir) un mécanisme de transport revient à se poser une question simple : « comment l'information arrive-t-elle au système de traitement ». Si la question semble simple, elle implique cependant un certain nombre de contraintes.

Sécurité

La première est bien évidemment la sécurité, ce à deux titres. Dans un premier temps, il y a les aspects d'authentification, d'intégrité et de confidentialité de l'information. Ces caractéristiques ne sont pas systématiquement intégrées aux protocoles de transport communément implémentés tels que syslog ou SNMP (or v3 bien entendu). En outre, le chiffrement, le déchiffrement et le calcul de *checksums* sont des opérations qui peuvent devenir lourdes dès que l'on traite un nombre important d'événements.

Dans un deuxième temps, le sens des flux est un élément important à prendre en compte dans la mesure où il n'est pas envisageable d'aller à l'encontre d'une politique de sécurité interdisant des flux entre certaines zones de sécurité. Dans d'autres cas, la récupération des informations peut nécessiter la définition de comptes spécifiques sur les systèmes (par exemple pour la récupération via FTP) voire la création de partages (Netbios, montages NFS etc.).

Si ces différents éléments peuvent dans certains cas être sécurisés, ils impliquent néanmoins un surcoût majeur en termes d'exploitation. Appliquer une politique de mot de passe cohérente (pas de mot de passe partagé, modification régulière, etc.) sur plusieurs dizaines, voire centaines, de systèmes est une entreprise considérable dont la complexité s'accroît avec l'hétérogénéité qui règne généralement dans les salles informatiques.

Performances

La deuxième contrainte à prendre en compte est l'impact sur les performances du réseau. Encore une fois, le transfert de volumes importants peut avoir un impact, en particulier aux points de convergence des flux d'information. Il est essentiel de garder à l'esprit deux éléments indispensables.

Tout d'abord le rôle d'un SOC est généralement d'automatiser le traitement d'un volume d'informations important. A défaut de quoi, un seul individu face à une console d'administration est largement suffisant. En outre certaines attaques vont générer un volume considérable de remontées d'information. Il peut s'agir d'un effet de bord de l'attaque ou d'une volonté délibérée de « noyer » le système. Et c'est en particulier dans ces cas extrêmes que le SOC doit être à même de jouer efficacement son rôle.

Équipements de sécurité

L'importance des informations émises par les équipements de sécurité est une évidence. Il s'agit en particulier des données transmises par les IDS, les IPS (Attention ! Le traitement ne sera pas identique, contrairement à ce que l'on peut croire), les équipements de filtrage, les anti-virus, etc. Il est également important d'inclure dans cette catégorie les outils d'analyse de vulnérabilités, dont les rapports pourront être liés aux informations « temps réel » afin de leur affecter une criticité adéquate et de réduire le taux de fausses alertes.

IDS et Scanners de vulnérabilités

Les outils de détection d'intrusions sont très utiles pour avoir une idée du niveau de trafic « hostile » qu'un réseau peut recevoir (les tentatives d'attaques) ou bien émettre (machines compromises, utilisateur malicieux). Les IDS ont tendance à générer un énorme volume de trafic – il n'est pas rare de voir plusieurs millions d'évènements IDS sur des réseaux de taille conséquente. Dans ce cas là, il est humainement impossible de distinguer les tentatives d'attaques des attaques réussies ou bien des faux positifs.

Pour gérer un tel volume, un outil d'analyse de logs va donc utiliser plusieurs méthodes pour extraire le signal du bruit :

Analyse statistique :

Par une analyse statistique, il est possible de déterminer que quelque chose ne va pas sur le réseau. Par exemple, si le nombre d'alertes d'IDS en semaine à 8h est habituellement de 10/mn alors qu'il est de 100/mn ce matin, alors une attaque est en train de se produire. Cette approche permet de détecter de grosses attaques, tel un ver en train de se propager ;

Corrélation avec des scanners de vulnérabilités :

Il est possible d'extraire du volume des attaques d'IDS la liste des attaques ayant eu de fortes chances de réussir en les corrélant avec les résultats d'un scan de vulnérabilités. Par exemple, si mon scan Nessus indique que la machine 172.20.16.33 est vulnérable à la faille UPnP de Windows et que mon Snort indique une tentative d'exploitation de cette faille, alors il y a de fortes chances que cette alerte soit une véritable intrusion – du moins beaucoup plus que le même paquet lancé en direction d'un OpenBSD 3.8. Cependant cette approche a ses propres limites : si le scan de vulnérabilités n'est pas fait très souvent et que le réseau est un réseau DHCP (ou bien que la machine cible soit en fait une instance VMWare), alors il est possible de créer de « faux négatifs » qui vont faire disparaître une attaque ayant réussie. Dans ce cas là, on préférera un scanner de vulnérabilités « temps réel » comme NeVO (Tenable) ou RNA (SourceFire) qui, par écoute du réseau, permet de connaître en temps réel la présence de certaines vulnérabilités, souvent moins efficacement qu'un scan actif mais souvent assez pour mieux qualifier les alertes des IDS.

Firewalls

Loguer toutes les requêtes d'un firewall peut générer un volume de trafic auprès duquel 2 ans de logs Snort sembleront succincts. Cependant, les logs de firewalls permettent à un outil de corrélation d'obtenir de nombreuses informations sur ce qu'il se passe dans le réseau. De plus, le firewall étant sans doute l'équipement de sécurité le plus déployé, il est aisé d'obtenir des logs des quatre coins du réseau sans déployer de nouveau programme.

En loguant les établissements de connexions ainsi que les paquets refusés, il va être possible pour l'analyseur de logs de déterminer certains changements dans le réseau par analyse statistique :

Un brusque changement de trafic :

Une soudaine hausse de tentatives de connexions allant de l'intérieur vers l'internet en direction de ports filtrés ou non va permettre de repérer des machines infectées par un ver. De même, une soudaine chute dans le volume échangé va permettre de repérer automatiquement le crash d'un routeur ou d'un serveur web interne.

Un brusque changement de comportement :

Une machine qui n'a que reçu des connexions de l'extérieur et qui n'a jamais établi de connexion vers l'extérieur qui tout à coup commence à établir des connexions HTTP est peut être une machine qui a été compromise. De même, une machine qui n'a jamais été qu'un client et qui devient un serveur est peut-être devenue un serveur FTP rempli DivX.

Enfin, stocker les logs d'un firewall dans un endroit centralisé permet de simplifier une analyse post-mortem : il devient possible de retracer les pas d'un pirate ayant compromis le réseau interne en quelques clics.

IPS et Antivirus

Les IPS et les antivirus ont pour rôle de bloquer les tentatives d'attaques et/ou d'infection. Leurs logs vont principalement permettre de repérer les machines infectées du réseau ou bien des utilisateurs facétieux en recherche d'émotions fortes en scannant le site web de la NASA.

Équipements d'infrastructure

Les équipements d'infrastructure sont les composants qui sont responsables du bon fonctionnement du système d'information. En termes de sécurité, il s'agit par exemple des serveurs d'annuaires, alors que du point de vue réseau, nous considérerons essentiellement routeurs et commutateurs.

Infrastructure de sécurité

Outre l'ensemble des composants intervenant dans l'authentification, cette catégorie intègre également les consoles d'administration des équipements de sécurité ainsi que les éléments du SOC lui-même. Dans ces derniers cas, la spécificité des événements par rapport à des événements intervenant sur un composant identique, mais hors du SOC, est essentiellement leur criticité.

Messages d'erreur

Le premier aspect est bien entendu l'ensemble des informations liées aux opérations d'authentification. Les erreurs d'authentification sont les événements qui viennent immédiatement à l'esprit, mais ils ne sont pas les seuls à fournir des informations pertinentes.

Les différents types de messages auxquels un SOC devrait faire attention sont les suivants :

Utilisateur refusé :

L'authentification est réussie mais l'utilisateur n'est pas (plus) autorisé à accéder à la ressource demandée. Ce message sera intéressant par exemple dans le cas d'une tentative d'abus d'un compte désactivé pour des raisons de sécurité (licenciement, décès, nommé RSSI etc.). La criticité dépendra généralement de la ressource cible et de l'utilisateur en question.

Utilisateur inexistant :

Le compte d'utilisateur n'est pas valide. La répétition de ce type d'événement est généralement caractéristique d'une tentative d'attaque par force brute des comptes d'utilisateurs. La criticité dépendra encore une fois de la ressource cible. Attention ! L'interception de ce type de message par l'attaquant lui donne un moyen de savoir si un compte existe ou non.

Erreur d'authentification :

Le classique. Néanmoins de nombreuses subtilités sont à prendre en compte, à savoir essentiellement : le compte utilisé, la source (local ou distant), la méthode et encore une fois la ressource « cible ». Ces éléments sont importants en particulier pour le calcul de la criticité.

Dans le cadre de la gestion de ces événements, leur nombre et le délai de répétition sont des éléments essentiels dans l'analyse des faits et la prise de décision. A partir de tels éléments la plupart des systèmes sont à même de générer des messages du type « erreur d'authentification multiple », synthétisant de manière triviale plusieurs messages d'erreur identiques reçus en un temps généralement assez court.

Sessions

Lorsque les opérations d'authentification sont centralisées, elles intègrent généralement des fonctionnalités d'*accounting*. Développées à l'origine pour des raisons de facturation, elles peuvent également fournir des informations précieuses concernant l'activité des utilisateurs et rendent possible la détection de comportements suspects.

Les informations intéressantes sont les suivantes.

Ouverture de session :

L'ouverture de sessions est un message simple, mais qui peut s'avérer particulièrement utile lorsqu'elle intervient dans un contexte anormal, que ce soit d'un point de vue temporel (compte d'une secrétaire à 3h du matin le dimanche) ou géographique (connexion depuis la Corée du nord ou la Tchétchénie). Les critères importants sont donc l'utilisateur, les classiques dates et heures (que l'on retrouve de toute façon dans tous les messages), la source et le service (ftp, ssh, intranet, etc.).

Fermeture de session :

Les informations concernant les fermetures de session sont importantes si elles sont corrélées avec les ouvertures de session. De cette manière, des anomalies telles qu'une succession de sessions très courtes ou inversement des sessions particulièrement longues peuvent être caractéristiques de comportements douteux tels que l'aspiration d'un site web protégé ou l'établissement (et le maintien) d'un tunnel.

Session timeout :

Ce type de message est essentiellement utilisé pour suivre le comportement des utilisateurs ayant accès à des comptes privilégiés. L'objectif étant de s'assurer que l'administrateur ne part pas prendre le café, la clope et la secrétaire en laissant sa session ssh ouverte en tant que *root* sur le SOA du domaine.

Gestion des comptes

La gestion centralisée des comptes utilisateurs et des droits qui leur sont associés est un avantage indiscutable pour la sécurité, tant d'un point de vue de l'administration que de la supervision. En effet, il devient possible de gérer à partir d'un seul point l'ensemble des droits attribués à un utilisateur ainsi que les accès que ce dernier a effectué sur les différentes ressources du système d'information.

Néanmoins, outre les autorisations d'accès, il est également particulièrement important de prendre en compte les messages concernant les opérations d'administration et/ou de maintenance sur ces comptes.

Les principaux types de message pertinents sont les suivants :

Modification d'un compte :

Il est évident que ce type de message doit être traité. Néanmoins, leur analyse et sa classification sont relativement complexes. Cette complexité vient du nombre important de paramètres qui entrent en considération. Le premier est le rôle de l'utilisateur dans la mesure où la modification d'utilisateurs sans droits particuliers n'a pas la même importance que la modification d'un compte administrateur. Il convient ensuite de qualifier le type de modification; il peut s'agir des droits, de la description, des horaires d'autorisation etc. Le troisième élément est la ressource « cible ». Dans le cas de droits, il convient de remonter l'information concernant les nouvelles autorisations (par exemple lecture/écriture) et la ressource « cible » (fichier, point de montage, etc.). Enfin, il est indispensable de fournir l'information concernant l'utilisateur qui est à l'origine de la modification.

Création d'un compte :

La problématique est la même que pour la modification d'un compte, à l'exception du fait que l'ensemble des paramètres du compte doit être transmis à l'issue de l'opération, ce qui peut poser des problématiques de transport et/ou de traitement et nécessiter parfois la corrélation de plusieurs messages.

Suppression d'un compte :

La nécessité et le type d'information à transmettre pour ce type de message sont évidents.

Outre la gestion unitaire des comptes, il est important également de prendre en compte les notions de « groupe » dans la mesure où les droits sont généralement appliqués de manière globale.

Il est donc nécessaire de prendre en compte au moins les événements suivants :

Modification d'un groupe :

La nature des informations remontées est la même que pour la modification d'un compte. La seule question à se poser est le type de message devant être affecté à un événement tel que la modification du groupe (ou de l'ensemble de groupes) auquel appartient un utilisateur. Il peut s'agir d'une modification du compte (appartenance à un groupe) ou du groupe (liste des utilisateurs).

Création d'un groupe :

La création d'un groupe nécessite de transmettre l'ensemble de ses caractéristiques et de ses membres. Le message peut, par conséquent, être relativement long.

Suppression d'un groupe :

Trivial.

Infrastructure réseau

Qu'il s'agisse de malversations internes, d'accès illégitime ou de propagation d'un code malicieux, les équipements d'infrastructure réseau sont à même de renvoyer de nombreuses informations pertinentes qui, selon les cas, permettront de compléter l'analyse d'un phénomène ou, dans d'autres cas, d'être les premiers à effectuer la détection. En outre, il est nécessaire de garder à l'esprit que ces équipements peuvent également être la cible de certaines attaques, ce, à des fins diverses et variées tels que le simple déni de service, le détournement d'information ou encore l'accès au réseau.

Problématiques génériques

Quelle que soit la nature des équipements, les informations concernant l'accès aux différentes interfaces d'administration (SNMP compris) et la modification de la configuration doivent impérativement être prises en compte.

Ces informations sont fournies par les événements suivant :

Accès au système :

Les différents messages liés à l'authentification (et erreurs associées) ainsi que la durée des sessions sont les mêmes que ceux identifiés précédemment.

Tentative de modification de la configuration :

Ce type de message est généralement remonté lors d'un échec lié à l'absence de droits permettant d'effectuer la modification. Cela peut être caractéristique d'un utilisateur malveillant ou d'une tentative à partir d'un compte « emprunté » de manière plus ou moins délicate... En l'occurrence, les éléments importants sont la source, l'utilisateur la date et l'heure. Idéalement, le détail des éléments sur lesquels porte la tentative de modification peut s'avérer également pertinent. Cependant, cette information est très rarement fournie dans de tels messages.

Modification de la configuration :

Il apparaît comme évident qu'une modification de la configuration doit impérativement être remontée au SOC. La problématique est le détail des opérations effectuées. Il est en effet relativement rare de voir les équipements fournir une information telle que « nouveau tunnel GRE créé », alors que dans certains cas cela pourrait être bien utile...

Il est donc généralement nécessaire de traiter ce type d'information en deux temps :

1. Réception d'un message précisant que la configuration a été modifiée.
2. Réception d'un message décrivant les modifications.

Le premier message sera généralement émis nativement par le système alors que le second sera la conséquence d'un script ou émis par un programme de contrôle d'intégrité à la *tripwire*.

Événements liés à l'accès au réseau

L'accès physique au réseau représente une menace directe pour la sécurité du système d'information. La criticité de cette menace est fonction du contexte, mais il n'est cependant pas concevable de laisser un accès au réseau filaire ou non, sans surveillance. Dès lors, il apparaît comme pertinent de prendre en compte les messages suivants :

Port up :

L'activation d'un port sur un équipement d'interconnexion est une information importante essentiellement dans deux contextes.

1. Équipement de *backbone*.
2. Équipement d'extrémité dédié aux serveurs.

Dans ces deux cas, il est évident qu'il s'agit d'équipements sur lesquels le changement du statut d'un port est un événement rare et nécessairement programmé, car il est normalement la conséquence d'un événement important tel que la mise en service d'un nouveau serveur ou l'ajout d'un équipement réseau. Enfin, dans les environnements « ultra-sécurisés », ce type d'événement peut également s'avérer utile sur les équipements d'extrémité. Il n'en reste pas moins que l'information devra être corrélée avec d'autres types de message (voir ci-dessous) dans la mesure où le moindre *reboot* ou démarrage matinal de chacun des PC de l'entreprise sera à l'origine de l'émission d'un tel message.

Nouvelle adresse MAC :

Ce type de message est généralement émis par des systèmes en écoute sur le réseau. Nous le traitons cependant ici, car il est la suite logique d'une intrusion physique sur le réseau. Afin de pouvoir qualifier rapidement la criticité du message, il est nécessaire d'avoir en parallèle l'adresse IP correspondant à l'adresse physique en question. Une duplication sera caractéristique d'une tentative d'*ARP poisoning* quand une modification sera (dans un plan d'adressage statique) caractéristique d'une connexion sauvage.

Association/Désassociation sur un AP :

Les nombreux articles sur les problématiques de la sécurité des réseaux WiFi devraient suffire à justifier la nécessité de la prise en compte de ce type de message. Les informations importantes sont l'adresse MAC et le BSSID. La gestion des messages de désassociation permet de prendre en compte la durée de la connexion ainsi que les tentatives de dénis de service. En cas d'utilisation d'EAP/802.1x, les informations d'authentification fournies par le serveur RADIUS seront également très pertinentes.

Les informations concernant l'accès au réseau sont rarement utilisables seules. Il est donc nécessaire de coupler leur traitement à différentes bases effectuant par exemple le lien entre adresse MAC et adresse IP ou adresse MAC et utilisateur, etc. De cette manière, il devient possible de fournir une réponse rapide à une action non autorisée, pourvu que les bases soient maintenues à jour...

Analyse du trafic

Enfin les derniers types de message pertinents concernant les équipements d'infrastructure réseau sont les messages liés au trafic, et ce, à deux titres : en cas de modification de la typologie de trafic et en cas d'injection de trafic risquant de compromettre l'intégrité de l'architecture.

La problématique de typologie correspond aux axes de charge, fréquence (paquets par seconde) et débit du trafic. La détection d'une anomalie dans un de ces axes est généralement caractéristique d'un comportement suspect. Il peut s'agir d'un déni de service (pic de fréquence et de débit), de la propagation d'un vers (pic de fréquence pour une charge identique) ou encore d'une tentative d'*ARP Flooding* (idem mais concernant un trafic de couche 2). Ce type d'alerte ne peut être généré qu'à l'issue de l'analyse d'information fournie par les équipements, que ce soit via un *polling* snmp ou une récupération de *netflow*. Dans tous les cas, aucune standardisation du traitement de ces messages n'est disponible à ce jour, ce qui signifie clairement que l'information issue de la corrélation dépendra des outils et scripts mis en place.

Néanmoins, ces derniers devraient remonter l'information suivante :

Anomalie de trafic :

Afin de permettre un traitement pertinent ce message doit contenir des informations concernant le type d'anomalie (fréquence et/ou débit et/ou charge), la nature du trafic (ARP, TCP/135, etc.), les éléments communs en termes de source et de destination (même adresse MAC source, même adresse IP destination, etc.) et l'écart en pourcentage par rapport à la valeur « normale ».

Enfin, la plupart des architectures réseau mettent en œuvre des protocoles de couche 2 ou 3 pour gérer la haute disponibilité, la gestion des routes ou encore fournir des informations de configuration. Les techniques d'exploitation de ces protocoles sont connues et il est important de pouvoir détecter toute tentative allant dans ce sens. Les attaques basées sur le volume seront traitées dans les anomalies. Néanmoins, il reste les attaques basées sur l'insertion et qui ne reposent pas nécessairement sur une variation caractéristique du trafic. On pensera à cette occasion aux tentatives de *hijacking* de la racine d'une architecture *spanning tree*, de l'usurpation d'un DHCP ou encore de l'injection de tables de routage.

Le type de messages à émettre sera donc le suivant :

Insertion de trafic :

Les éléments importants sont les informations de niveau 2 et 3 concernant la source et la cible. Tous ces éléments sont indispensables dans la mesure où la plupart de ces informations sont *spoofées* ou sont des adresses de *broadcast*. Néanmoins, la corrélation avec d'autres types de message, tels que les messages d'accès au réseau, permettent généralement de faire le tri et d'identifier la source malicieuse. Les autres informations pertinentes sont le protocole attaqué et le type d'information insérée.

Il ressort assez évidemment que les messages de ce type sont les plus difficiles à générer et à intégrer de manière homogène dans les outils de traitement du SOC. Ils sont néanmoins d'une importance capitale pour l'investigation d'incidents passés ou en cours sur un réseau interne.

ROW (Rest of the World)

La récupération d'informations provenant des éléments de sécurité et d'infrastructure est un élément indispensable au bon fonctionnement d'un SOC. Cependant, cela reste souvent insuffisant pour construire efficacement le schéma d'une intrusion et décider d'une action, que ce soit en temps réel ou en différé. Un exemple simple est la détection d'un *exploit* (remontée par un IDS) à destination d'un serveur. Et après ? L'attaque a-t-elle été réussie ? Si oui, quelles opérations ont été effectuées ? La même question se pose lors de l'authentification d'un administrateur à trois heures du matin alors qu'aucun dysfonctionnement ne justifiait cette connexion. Il est donc nécessaire de récupérer les informations émises par les ressources que l'entreprise veut protéger.

Dans la mesure où il serait parfaitement illusoire de vouloir classer ces différents événements, nous les fournissons tels quels.

Identification de la cible :

Certaines applications sont à même d'identifier une connexion dont le seul objectif est l'obtention d'informations telle que la version. Les critères importants dans ces cas-là sont l'application cible et la source. Ils permettront de reconstruire le schéma d'une tentative d'intrusion dans la mesure où ce type de message peut être corrélé avec des messages de scan de ports et/ou de tentative de *fingerpinting* d'OS, puis de lancement d'*exploit*.

Récupération massive d'information :

Selon les applications, il est possible de récupérer un volume conséquent d'informations pertinentes. C'est le cas des serveurs DNS ou des serveurs SMTP, par exemple. Ainsi toute tentative de dump de zone ou de lancement d'une commande `EXPN` est un événement important qui doit être remonté par l'application au SOC. Par exemple, le message suivant : `Sep 27 17:20:46 lab1 sshd[17942]: scanned from 192.168.202.104 with SSH-1.0-SSH_Version_Mapper. Don't panic.` devra fournir l'application (ssh) et la source (192.168.202.104).

Effacement des logs :

La nécessité de remonter ce type d'information est une évidence. En dehors d'une opération de maintenance programmée, elle caractérise probablement une intrusion fructueuse suivie d'une opération peu subtile de dissimulation des traces. Il y a peu d'information importante à retenir dans ce cas, hormis peut-être le nom utilisateur ayant effectué l'opération, lequel est généralement `root` ou `Administrator`, aïe !

Arrêt et (re)démarrage d'un service :

Dans le même ordre d'idées, une des opérations communément effectuée à l'issue d'une intrusion est l'arrêt des services d'enregistrement (syslogd, klogd) pendant la durée des opérations de corruption de ces derniers. Plus rarement, il peut arriver qu'un exploit crashe le service cible ou du moins l'un des processus. Qu'il soit ensuite relancé par un *watchdog* ou l'exploit lui-

même, l'information concernant son redémarrage peut être particulièrement pertinente. Dans les deux cas, il est important d'être à même de connaître le service concerné et la durée de l'interruption.

Corruption/Modification de fichier :

Généralement remontés par des mécanismes de contrôle d'intégrité, parfois intégrés aux anti-virus. Ce type d'information est particulièrement importante quand il s'agit de fichiers du système (librairies, binaires voire noyau) ou de fichiers de configuration critiques tels que la liste des utilisateurs et des *hashs* de leurs mots de passe, la liste des utilisateurs autorisés à se connecter à un service, etc. Il est alors indispensable de fournir l'information concernant la ressource affectée et le type de modification détectée. Idéalement une fenêtre temporelle au cours de laquelle la modification a été effectuée (depuis la dernière vérification) serait également une information pertinente et permettrait rapidement de faire le lien avec d'autres événements intervenus au cours de la même période.

Mise à jour :

L'installation de patches est une information essentielle dans l'évaluation de la gravité d'un incident. En effet la détection de comportements douteux sur une machine dont la dernière mise à jour date de plusieurs mois n'est pas nécessairement une surprise. En revanche, si la mise à jour date de la veille, c'est qu'il y a quelque chose de très méchant qui se balade dans le quartier. Dans un autre ordre d'idée, le suivi des opérations de mise à jour des solutions anti-(virale|*spyware*|*spam*|.*) aussi bien que des composants du système (OS et application) ne peut se faire sans la collecte de ce type d'événement. Il est donc nécessaire de récupérer la date de la mise à jour, la ressource affectée, sa version et le statut de l'opération (succès ou échec).

Il convient bien sûr de rajouter à ces événements les informations liées à l'authentification locale, aux privilèges et aux droits lorsque ceux-ci sont gérés localement au niveau du système d'exploitation et/ou des applications.

Conclusion

Fournir des informations pertinentes pour permettre au SOC de faire convenablement son travail est une problématique qui va bien au-delà de la simple centralisation des logs de quelques firewalls. Il est cependant encore relativement rare de voir un centre de supervision intégrer à ses données l'analyse complète des informations potentiellement pertinentes. Cet état de fait est lié à un certain nombre de phénomènes.

Tout d'abord, tous les SOC n'ont pas pour mission de superviser l'ensemble des problématiques de sécurité : Dans certains cas, un SOC n'a pour fonction que de surveiller l'activité virale, dans d'autres, l'accès aux ressources critiques, etc. La deuxième raison est généralement le manque d'outils appropriés pour effectuer une telle collecte. Enfin, l'organisation, que ce soit au niveau de l'urbanisation des données, des rapports générés (et surtout de leurs destinataires) ou, enfin, des ressources humaines afin de réagir de manière adéquate, cette organisation donc, est rarement mise en place de manière globale à l'entreprise, car, comme chacun sait, « la politique en entreprise est à la sécurité du SI ce qu'une braguette coincée est à une envie de pisser » [HB].

Référence

[HB] – HB (hb.rstack.org), 3 Juin 2005, 4h15, SSTIC 2005

Des outils libres pour superviser la sécurité

Philippe Lagadec – DGA/CELAR
philippe.lagadec@dga.defense.gouv.fr
Jean-François Suret – DGA/CELAR
jf.suret@gmail.com
Samuel Dralet
zg@kernsh.org

Les systèmes d'exploitation et les équipements réseau actuels fournissent tous des mécanismes de base pour surveiller les événements qui concernent la sécurité (syslog, eventlog, SNMP,...). Cependant ces mécanismes seuls sont insuffisants pour superviser de façon globale et efficace la sécurité d'un système d'information hétérogène, alors qu'il s'agit d'un besoin croissant pour la plupart des entreprises et administrations.

De nombreux outils commerciaux, gratuits ou open source existent aujourd'hui pour répondre à ce besoin (cf. [LOGAN]), bien que leur mise en œuvre ne soit pas encore « plug and play ».

Cet article a pour but de montrer qu'il est tout à fait possible de concevoir une infrastructure complète de supervision de la sécurité d'un système hétérogène à base d'outils libres. Pour cela, les diverses fonctions nécessaires sont détaillées et illustrées à l'aide d'exemples concrets.

Hypothèses et besoins

La plupart des systèmes d'information actuels dans une entreprise ou une administration sont relativement hétérogènes. On y trouve généralement des postes clients ou serveurs Windows (NT4 à 2003, voire Windows 9x), des postes Unix (Linux, *BSD, Solaris, AIX,...), ainsi que divers matériels réseau (routeurs et commutateurs Cisco, 3Com, ...) et autres « appliances » (pare-feu, *proxies*, IDS/IPS, ...). Sur les machines, on peut trouver de nombreuses applications aussi diverses que variées : serveurs web, serveurs de fichiers, de messagerie, bases de données, antivirus, applications métier, etc. Et pour simplifier les choses, certains systèmes d'information sont répartis sur plusieurs sites géographiques, reliés entre eux par des liaisons réseau plus ou moins performantes.

La sécurité globale du système d'information repose sur tous ces éléments à la fois, aucun n'est à négliger. Chacun de ces matériels et logiciels peut générer et enregistrer régulièrement des événements, afin de tracer les actions normales ou anormales survenant lors de son fonctionnement.

Superviser la sécurité du système d'information nécessite l'accès à tous ces événements ou plus précisément à ceux qui sont les plus significatifs. L'objectif final est de permettre à un administrateur de suivre de façon globale l'évolution de son système d'information, ceci afin de réagir rapidement en cas de problème ou d'action malveillante au lieu de s'en rendre compte a posteriori. L'expérience montre que les systèmes de détection d'intrusion classiques sont utiles, mais insuffisants dans cette tâche. Une infrastructure de centralisation et de supervision des événements sécurité est donc nécessaire.

Définitions : événement, log

Dans la suite de cet article, on définit un « événement » (ou « event » en anglais) comme un message de texte décrivant une action ayant eu lieu sur un système informatique. Ce message est accompagné de la date et l'heure, la machine concernée, ainsi que diverses informations éventuelles : système, application, utilisateur, niveau de criticité,...

Voici un exemple d'événement (purement fictif) :

```
09-11-01 17:05  ERROR   ftp[922] : unable to connect to server ftp.rstack.org.
```

Cet événement est généralement stocké dans une ligne d'un fichier texte, dans un fichier binaire ou bien une base de données. Ce fichier est appelé journal d'événements, « log » ou encore « logfile ».

Tour de Babel

Dans un système classique, chaque matériel ou logiciel stocke localement ses événements propres, sous une forme qui n'est pas forcément standardisée ou interopérable (EventLog sous Windows, syslog sous Unix, etc.).

Le premier besoin est donc de centraliser tous les événements de façon homogène, afin de pouvoir les exploiter globalement.

Trop de logs tue les logs

En général, aucun administrateur normalement constitué n'est capable d'assurer la supervision complète d'un système dès que sa taille dépasse 3 ou 4 machines. Les événements générés sont beaucoup trop nombreux et souvent trop parcellaires ou trop ésotériques pour qu'un humain standard puisse avoir une vision globale et claire de la sécurité du système.

Le second besoin est donc de réduire la masse d'information produite pour ne conserver que les événements importants, tout en corrélant certaines données pour rendre les événements plus significatifs.

Inconnu à cette adresse

Le nombre d'événements différents qu'un système d'information classique peut générer est considérable (il suffit de lire un journal d'événements Windows pour s'en rendre compte) et la plupart des documentations de systèmes d'exploitation ou de logiciels ne décrivent pas la liste exhaustive des événements possibles. Pour arranger les choses, l'application d'un correctif ou d'une mise à jour peut très bien modifier le libellé de certains événements, ajouter ou supprimer des évènements.

On prendra donc comme hypothèse qu'à un moment donné on ne peut connaître qu'un sous-ensemble limité des événements qui peuvent être générés. Un événement dont le libellé est inconnu peut survenir à tout instant. On ne peut se contenter de traiter uniquement une liste d'événements connus, car un événement

inconnu jusqu'ici peut être capital pour la sécurité. Ce type d'événement qui nous intéresse a justement pour habitude de se produire rarement...

Supervision

Pour qu'un administrateur puisse superviser la sécurité, il doit pouvoir consulter le journal global des événements à l'aide d'une interface adaptée. Par exemple les événements les plus critiques doivent apparaître clairement, il doit être possible de filtrer l'affichage suivant divers critères, faire des recherches, camoufler des événements jugés secondaires, marquer les événements traités, etc.

Pour un système de taille importante supervisé par plusieurs personnes, il est souvent nécessaire d'avoir une interface plus évoluée qui permet de traiter et de suivre plus efficacement les événements traités par chaque intervenant, par exemple à l'aide de « tickets d'incidents ».

Deux projets à base d'outils libres

Pour illustrer concrètement la mise en œuvre des méthodes et des outils présentés par la suite, deux projets complémentaires sont mentionnés : Log3C et SFS.

Log3C

Log3C (Création, Centralisation et Corrélation de logs) est une expérimentation du CELAR qui a consisté à construire un réseau hétérogène représentatif de systèmes d'information réels (clients et serveurs Windows, Linux, routeurs, pare-feu, serveurs web, messagerie, base de données, ...) et à bâtir un système complet de supervision de la sécurité à base d'outils libres, notamment Prelude, SEC, syslog-ng, NTsyslog, logger, OpenVPN et Nagios.

La méthode employée est la suivante :

- Tous les événements générés sont centralisés et traités de façon homogène.
- Les événements normaux et répétitifs qui ne sont pas pertinents pour la supervision de la sécurité sont éliminés grâce à une première liste de règles.
- Les événements connus pour être importants sont pris en compte, en remaniant éventuellement leur libellé pour les rendre plus compréhensibles et en les corrélant lorsque c'est possible. Cela est effectué grâce à une seconde liste de règles.
- Les événements restants, qui ne correspondent à aucune règle, sont remontés en tant qu'événements inconnus. L'administrateur doit alors compléter les 2 listes de règles précédentes pour qu'ils soient correctement reconnus par la suite.
- Ces événements importants et inconnus sont affichés en temps réel dans une interface web.
- Au final, le système de supervision de la sécurité est complété et affiné de façon itérative. L'avantage principal de cette méthode est que tout nouvel événement sera pris en compte. On ne se contente pas de superviser des événements connus.

SFS : Centralisation de logs à base de SSH

Dans cet article, la solution adoptée par Samuel est nommée SFS, pour « Supervision Furtive Sécurisée ». SFS rassemble quasiment les mêmes spécificités que Log3C : centralisation, nettoyage des évènements inutiles et exportation. Cependant la solution a été pensée différemment. Alors que Log3C a visé une approche globale permettant de corréler les événements, SFS est développé dans le but de détecter en priorité d'éventuelles intrusions et de remonter les évènements le plus efficacement possible.

On suppose qu'à tout moment le système peut être introduit par un pirate, par conséquent il est nécessaire de pouvoir récupérer les logs avant que le pirate ne puisse les modifier.

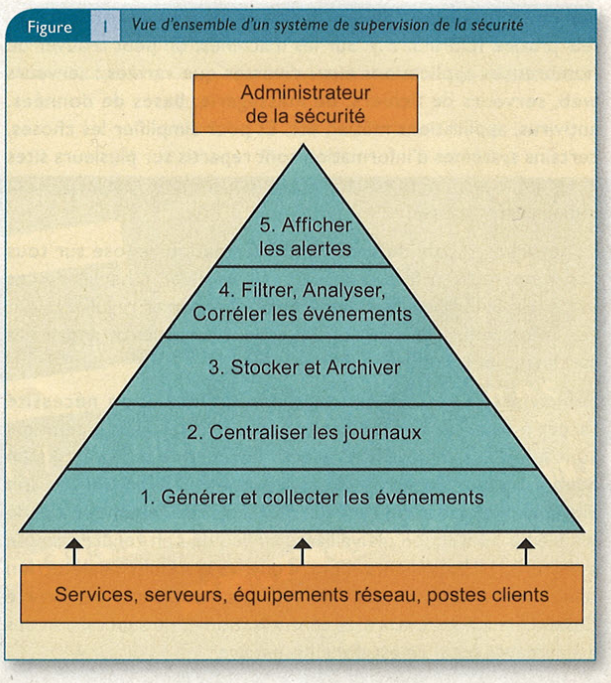
L'outil se décompose en 3 parties :

- L'interception des évènements : elle est faite de manière furtive avant que les évènements ne soient enregistrés.
- Le traitement des évènements : il est fait à base d'expressions régulières. Les évènements normaux et répétitifs sont supprimés, seuls restent ceux présentant un danger ou ne correspondant à aucune règle. Dans le cas des évènements inconnus, les règles sont alors complétées et affinées de manière à ce qu'ils soient reconnus la fois suivante.
- Le transport des évènements : une fois qu'ils sont traités, les évènements sont chiffrés et transportés via un tunnel SSH vers un point central, duquel ils sont renvoyés par SMTP vers des boîtes aux lettres dédiées.

Les fonctions d'un système de supervision de la sécurité

Un système de supervision de la sécurité peut être découpé en 5 fonctions principales que nous allons détailler.

Figure 1 *Vue d'ensemble d'un système de supervision de la sécurité*



Fonction 1 : Générer et collecter les évènements

Rares sont les systèmes d'information basés sur un parc d'ordinateurs homogène. Or les systèmes d'exploitation et équipements réseau n'utilisent pas tous les mêmes méthodes et protocoles pour gérer les journaux d'événements.

Windows

Les systèmes d'exploitation Windows (NT, 2000, XP et 2003) utilisent une méthode propriétaire pour stocker les informations dans des journaux séparés selon les thématiques suivantes : Applications, Sécurité, Service d'annuaire, Service de réplication de fichiers, Serveur DNS et Système. Ces journaux d'événements (appelés « eventlog ») sont stockés en local sur chaque poste dans des fichiers binaires avec un format propriétaire.

Après une installation par défaut, aucun événement n'est inscrit dans le journal Sécurité. Il est donc nécessaire d'activer manuellement la stratégie d'audit pour journaliser tous les événements importants comme les connexions/déconnexions d'utilisateurs et les modifications dans la base des utilisateurs.

Unix, Linux et autres BSD

Sur la plupart des systèmes Unix, les logs sont gérés par le démon syslogd. Ce démon permet, à l'aide de son fichier de configuration `/etc/syslog.conf`, de personnaliser la gestion des logs.

Il est ainsi possible d'enregistrer toutes les authentifications réalisées par le système, de surveiller ce qui se passe au niveau de son noyau, de rediriger absolument tous les logs vers un terminal, etc., le tout en mode texte (donc facilement lisible) :

```
$ tail /var/log/kern.log
Sep 23 10:25:04 localhost kernel: hdb: command error: status=0x51 { DriveReady
SeekComplete Error }
Sep 23 10:25:04 localhost kernel: hdb: command error: error=0x51 {
IllegalLengthIndication LastFailedSense=0x05 }
Sep 23 10:25:04 localhost kernel: ide: failed opcode was: unknown
Sep 23 10:25:04 localhost kernel: end_request: I/O error, dev hdb, sector 64
Sep 23 10:25:04 localhost kernel: isofs_fill_super: bread failed, dev=hdb,
iso_blknum=16, block=16
Sep 23 11:09:44 localhost kernel: kjournald starting.  Commit interval 5 seconds
Sep 23 11:09:44 localhost kernel: EXT3-fs warning: maximal mount count reached,
running e2fsck is recommended
Sep 23 11:09:44 localhost kernel: EXT3 FS on loop0, internal journal
Sep 23 11:09:44 localhost kernel: EXT3-fs: mounted filesystem with ordered data
mode.
Sep 23 13:22:05 localhost kernel: spurious 8259A interrupt: IRQ7.
```

En plus du démon syslogd, généralement trois fichiers supplémentaires fournissent de précieuses informations : `/var/log/wtmp`, `/var/log/lastlog` et `/var/run/utmp`. Ces fichiers sont maintenus par les programmes init, login et getty. Le premier, `/var/log/wtmp`, enregistre les connexions et déconnexions au système et peut être visualisé grâce à la commande `/usr/bin/last`. Le second, `/var/log/lastlog`, contient un historique des dernières connexions et peut être visualisé grâce à la commande `/usr/bin/lastlog`. Le dernier, `/var/run/utmp`, permet de voir qui est connecté sur le système et peut être lu avec la commande `/usr/bin/w`.

Ces fichiers sont organisés sous forme d'enregistrements d'une structure et sont par conséquent au format binaire.

Applications

Il existe aussi des applications comme Oracle ou Apache qui n'utilisent pas ces mécanismes de gestion des événements propres aux systèmes d'exploitation. Ces serveurs stockent leurs informations dans des fichiers textes ou des bases de données sous des formats propriétaires.

Équipements réseau

La plupart des équipements réseau (routeurs, commutateurs,...) sont capables de journaliser des événements grâce à syslog ou SNMP, cependant chaque constructeur peut avoir ses spécificités.

Prenons par exemple le cas répandu des routeurs Cisco avec des *access-lists* de filtrage. Il est possible d'émettre des événements vers un serveur via SNMP-trap, mais dans ce cas uniquement les événements concernant le fonctionnement du routeur seront émis (interfaces up/down, etc.). Seul syslog permet de journaliser des événements concernant la sécurité et le filtrage. De plus, il est nécessaire de spécifier chaque ligne d'access-list devant générer des événements. Voici un extrait de configuration de routeur Cisco qui permet de journaliser via syslog tous les paquets filtrés par une access-list, grâce au mot clé « log » :

```
access-list 100 permit tcp 192.168.1.0 0.0.0.255 any eq 80
access-list 100 permit tcp 192.168.1.0 0.0.0.255 any eq 443
[…]
access-list 100 deny tcp any range 0 65535 any range 0 65535 log
access-list 100 deny udp any range 0 65535 any range 0 65535 log
access-list 100 deny ip any any log
```

On peut noter que les 3 dernières lignes sont nécessaires pour que le routeur indique les numéros de ports TCP ou UDP des paquets filtrés dans les événements syslog.

Événements nominatifs

Pour une supervision efficace, chaque événement journalisé doit contenir le maximum de détails sur la machine, le processus, l'utilisateur,... bien sûr en respectant toujours les principes énoncés dans le premier article de ce dossier. Pour déterminer quel utilisateur réel est à l'origine d'un événement, il peut être nécessaire d'adapter la configuration des machines. Par exemple sous Unix, il peut être envisagé d'interdire la connexion directe sous un compte *root* (anonyme par nature), en autorisant uniquement le *login* pour des comptes utilisateurs non privilégiés. Toute action d'administration nécessite alors l'utilisation des commandes `su` ou `sudo`, après une connexion nominative bien visible dans les journaux d'événements.

Collecter les événements de manière furtive

Dans le cas d'une intrusion, se fier aux fichiers de logs des systèmes ou applications peut être une erreur. Les emplacements des fichiers sont connus de tous, rien n'empêche le pirate lors d'une intrusion (c'est d'ailleurs ce qu'il fait en premier), d'aller les modifier pour effacer ses traces.

Pour assurer l'authenticité des logs de notre système, deux conditions s'imposent :

- Les logs ne doivent pas être écrits aux emplacements par défaut (`/var/log/syslog` par exemple). Soit, il faut les rediriger vers un fichier caché et dans ce cas-là, le mieux est de chiffrer

les données, soit, idéalement, vous envoyez vos données avant même qu'elles ne soient écrites sur le disque.

Le système doit être furtif, invisible aux yeux du pirate sans quoi la solution n'a aucun intérêt.

Pour remplir ces deux conditions, il n'y a pas meilleur moyen que d'utiliser des techniques de porte dérobée. Le pirate est en quelque sorte piraté.

Focalisons-nous sur Linux car c'est sur cette plate-forme qu'a été développée SFS. Concrètement et d'une manière générale, la solution la plus simple pour intercepter un événement est de `hook()er` l'appel système du démon qui reçoit (`recv()`) ou qui va écrire (`write()`) l'évènement (par exemple pour le démon syslogd, ce sera `recv()`). Vous avez alors le choix de réaliser l'outil en mode noyau ou en mode utilisateur.

En mode noyau, il est possible par exemple de modifier la table des appels système pour utiliser votre propre appel système `recv()` qui fera évidemment ce que vous lui demander de faire. On peut aussi détourner des outils de leurs fonctions initiales tel que Uberlogger [UBER] ou Sebek [SEBEK] destinés tous les deux au monde des *honeypots*. Sebek par exemple surveille un système par interception d'appels système. Il vérifie entre autres la création de processus, l'activité des processus, les données échangées et traitées, etc. Il serait tout à fait adéquat (avec peut-être quelques petites modifications) pour un rôle de supervision. La présentation [EUROSEC] décrit précisément ce cas de figure.

En mode utilisateur, l'idéal est de faire appel à des techniques d'injection ELF, c'est facile à développer avec la librairie `elfsh` [ELFSH] et c'est plutôt portable (Linux, Solaris, FreeBSD, etc.). Un exemple d'injection appelée « ALTPLT » permet de modifier la section .plt des binaires ELF pour rediriger certains appels de fonctions vers le code injecté. De cette manière, vous pouvez modifier la fonction `recv()` ou `write()` du binaire que vous souhaitez. L'inconvénient des solutions en mode utilisateur est qu'il peut être nécessaire de développer une porte dérobée pour chaque application qui génère des évènements. On peut certes charger une librairie globale à l'aide du fichier `/etc/ld.preload`, mais la technique n'est pas très furtive. Pour ceux qui ne connaissent pas ce fichier, il permet de configurer la variable d'environnement `LD_PRELOAD` pour chaque programme qui va être lancé. Cette variable permet de référencer des bibliothèques dynamiques qui seront chargées avant toutes celles utilisées par le binaire. Les symboles exportés par les librairies de `LD_PRELOAD` remplaceront alors ceux des bibliothèques standards appelées par le binaire. Une solution plus radicale est de *backdoorer* la `libc`, librairie avec laquelle chaque binaire est compilé. Du coup, si nous modifions les fonctions concernées de cette librairie, ces modifications s'appliquent alors à chaque programme qui utilise ces fonctions.

En résumé, il existe moult techniques de *backdoor* pour se construire un outil de collecte de logs dit « furtif » à condition qu'il soit développé par vous. Des outils comme Sebek sont inefficaces. Ils sont connus de tous, donc on sait comment les détecter. Rien n'empêche alors un pirate d'utiliser un *shellcode* qui détecte la présence ou non de Sebek dans son *exploit*, de manière à lancer un shell en toute sécurité.

Fonction 2 : Centraliser les journaux

La centralisation peut sembler simple à mettre en œuvre. Il suffit de regrouper tous les journaux d'événements sur un serveur commun. Pourtant, il existe plusieurs points concernant la centralisation sur lesquels on se doit de rester prudent !

Windows

Sous Windows, aucun mécanisme de centralisation natif n'est proposé pour les journaux d'événements. Le seul outil fourni en standard est un « observateur d'événements » (« EventViewer » en anglais), qui offre des fonctionnalités rudimentaires pour accéder via le réseau à un journal d'événements d'une seule machine à la fois et pour le filtrer suivant quelques critères.

Il est possible d'envisager la centralisation d'événements Windows grâce aux produit SMS et au nouveau MOM 2005 (cf. [MOM]), cependant cela ne pourra concerner a priori que les produits Microsoft.

Syslog

Aujourd'hui, le seul protocole standard et suffisamment répandu pour centraliser les événements de tous les systèmes est syslog (cf. [RFC3164]). La plupart des équipements réseau et des systèmes d'exploitation Unix l'implémentent. Ainsi syslog apparaît comme un système simple à utiliser pour mettre les différentes informations au même format.

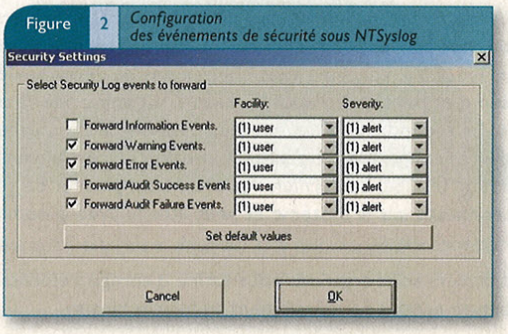
On pourrait aussi parler de SNMP qui sert énormément à la supervision des équipements réseau, cependant SNMP n'a pas été imaginé pour répondre aux besoins de sécurité des systèmes et il serait plus compliqué que syslog à utiliser dans ce cadre.

La décision d'utiliser syslog pour gérer les journaux d'événements implique de mettre en place des mécanismes de traduction des journaux d'événements Windows ou de fichiers textes vers syslog.

Journaux d'événements Windows vers syslog

Pour procéder à l'envoi des journaux d'événements Windows vers un serveur syslog, on dispose de plusieurs outils *open source*. NTSyslog (cf. [NTSL]) est à ce jour l'outil le plus connu, il permet une configuration avancée des informations que l'on veut envoyer au serveur central syslog. Il existe aussi d'autres outils ayant les mêmes fonctions, tel que Snare (cf. [SNARE]) ou Eventlog to Syslog. Tous ces outils sont basés sur des interfaces graphiques et sont très simples à utiliser, alors pourquoi s'en priver ?

Figure 2 *Configuration des événements de sécurité sous NTSyslog*



Fichiers textes vers syslog

Concernant les fichiers textes, il existe sous GNU/Linux une commande qui se nomme `logger` et qui permet de réagir dès la modification d'un fichier pour envoyer les nouvelles informations en syslog (cf. [LOGGER]). Cette commande est par exemple utilisée lorsque l'on souhaite enregistrer les informations des demandes d'accès d'Apache en syslog. Pour mettre en place cette méthode, il suffit de modifier la configuration de apache de la façon suivante :

```
CustomLog "| /usr/bin/logger -t 'apache_access_log'" combined
```

Ainsi toutes les requêtes d'accès au serveur apache seront envoyées au démon syslog avec comme identifiant le tag `apache_access_log`. Ceci est un exemple d'utilisation et ne doit pas être utilisé dans tous les cas, en effet les demandes d'accès sont très nombreuses sur un serveur web et cela génère énormément de données. Il peut donc être intéressant de procéder à un premier filtrage pour n'envoyer que les lignes utiles.

Sous Windows il n'existe pas d'outil comparable, cependant il est très simple de développer un programme pour réaliser la même fonction (un programme de ce type a été écrit en Python dans le cadre de Log3C). Il est aussi possible de trouver des outils tel que Adiscon logger [ADISCON] qui propose les mêmes fonctionnalités que la commande Unix logger, mais sous Windows.

Centraliser l'information de façon sécurisée

La protection des échanges d'information concernant la sécurité est un point critique. Cela concerne la disponibilité, la confidentialité, l'intégrité et l'authenticité des transmissions des journaux d'événements.

Il faut impérativement que les informations arrivent au destinataire. Or syslog repose sur UDP qui n'apporte aucune garantie de transmission. C'est pourquoi il est intéressant d'utiliser des systèmes syslog alternatifs comme par exemple syslog-ng (cf. [SLNG]). Celui-ci apporte les mêmes fonctions que syslog avec une compatibilité ascendante, tout en ajoutant la possibilité d'utiliser TCP et en proposant des filtrages avancés.

D'autre part, il faut éviter qu'un individu malveillant puisse générer de faux événements, modifier les événements transmis à la volée ou bien écouter des événements contenant des informations potentiellement confidentielles. Pour cela, la solution classique consiste à chiffrer et signer les données. Or pour le moment syslog et syslog-ng ne proposent pas de mécanisme de chiffrement, même si l'implémentation future de cette fonction est prévue. Il faut donc passer par des moyens de sécurisation complémentaires. L'utilisation d'IPSec étant contraignante pour réaliser de simples tunnels chiffrés, il est préférable de se reporter à des solutions plus légères telles que SSH pour sécuriser les flux TCP (cf. [SSH] et [OPENSSH]) ou OpenVPN qui permet de créer des tunnels chiffrés et authentifiés pour des flux TCP ou UDP (cf. [OPENVPN]).

Si l'on dispose d'un budget suffisant, il est aussi possible d'envisager le déploiement d'un second réseau physique réservé aux flux d'administration, afin d'éviter le recours au chiffrement. Cependant cette solution est moins satisfaisante en cas de compromission d'une machine.

SSH

SFS utilise justement SSH (OpenSSH pour Linux plus précisément), qui présente l'avantage d'être porté sur un maximum de plates-formes y compris Windows. La problématique est la suivante : pouvoir à partir d'une machine unique récupérer les événements de chaque machine à surveiller en toute transparence.

La solution retenue est une architecture à base de clés et de ssh-agent le tout géré avec keychain [KEYCHAIN]. C'est un outil qui permet d'utiliser un ssh-agent par système et par utilisateur et non pas par *login session*. ssh-agent est un démon utilisé dans le seul but de mettre en cache nos clés privées déchiffrées. SSH peut communiquer avec ssh-agent, lui permettant d'acquérir nos clés privées sans avoir à taper un *password* ou une *passphrase* à chaque nouvelle connexion. Seulement, à chaque fois que nous lançons ssh-agent (par l'intermédiaire d'une session login ou à la main), nous sommes obligés d'utiliser ssh-add pour ajouter des clés, nécessitant alors de taper sa passphrase. Keychain résout le problème. Du coup un seul ssh-add suffit et un seul ssh-agent tourne. Keychain optimise le processus ssh-add en essayant seulement d'ajouter les clés privées qui ne sont pas déjà dans le cache de ssh-agent. Le seul problème avec keychain est qu'il n'est pas porté sur Windows. Vous devez donc vous contenter du ssh-agent seul sur ce système qui est en fait remplacé par un programme appelé « Pageant ». Des tests pour essayer de se rapprocher le plus d'une architecture Unix sont à effectuer.

Pour conclure, outre le fait qu'elle centralise les logs, une telle architecture permet à partir de scripts d'administration de mettre à jour facilement le système de traitement des événements sur chaque machine. OpenSSH par exemple offre la possibilité de lancer une commande sur la machine distante plutôt que d'ouvrir une session. Il paraît aussi évident que la machine qui centralise les logs sera la plus sensible. Si un pirate la compromet, il aura accès à toutes les autres machines du réseau grâce au ssh-agent qui tourne. Dernier point à définir, la création des clés. Qui les crée et à quel moment ? Il faut savoir qu'OpenSSH a des fonctionnalités de détection de *spoofing*. C'est-à-dire qu'à chaque modification de clés par exemple, OpenSSH nous avertit. Est-il donc nécessaire de changer la paire de clés privée/publique périodiquement ?

Synchroniser les horloges des machines

Un des points à ne pas oublier, lorsque l'on parle de centralisation des journaux, concerne les instants où sont sauvegardées les informations. En effet pour pouvoir faire un traitement ultérieur et notamment de la corrélation, il faut que les événements soient sauvegardés dans un ordre cohérent. Pour cela, il existe des serveurs de temps NTP ou SNTP qui permettent de synchroniser les horloges des équipements réseau et des serveurs (cf. [RFC1305] et [RFC2030]). Lorsque le système d'information est réparti sur plusieurs fuseaux horaires, il faut également penser à choisir un fuseau de référence.

Ce mécanisme se doit lui aussi d'être sécurisé pour éviter qu'un individu extérieur puisse modifier les dates des serveurs et équipements, impactant ainsi la cohérence des logs. Pour cela, l'implémentation standard de NTP propose l'authentification des serveurs par clés MD5.

Fonction 3 : Stocker et archiver

Tout au long du processus de remontée d'alerte, les données sont manipulées à différents niveaux du système d'information. Ces données transitent des équipements et machines vers le ou les serveurs de centralisation. Or pour assurer un traitement ultérieur de ces données en cas d'attaque importante, il est nécessaire de protéger les informations et de les archiver aussi bien à leur source qu'au niveau des serveurs de centralisation.

Enregistrer l'information ce n'est pas simplement stocker les données. Il faut que les données soient protégées, pour éviter qu'un individu extérieur puisse les modifier. Plusieurs mécanismes sont à mettre en place : protection des droits d'accès grâce au système de fichiers (seul les éléments générant des événements doivent pouvoir modifier les fichiers), rotation des journaux d'événements pour sauvegarde, s'assurer que les disques durs ne sont pas saturés,...

Lorsqu'on utilise un service comme syslog, les informations sont stockées dans des fichiers textes. Ces fichiers peuvent être séparés selon plusieurs critères tels que la sévérité du message ou sa provenance. Le problème est que l'on ne peut pas garder ces informations « ad vitam ». En effet, les fichiers risqueraient de prendre trop de place, de saturer l'espace disque, et il deviendrait fastidieux de faire des recherches. L'outil Logrotate est alors utilisé pour réaliser la « rotation » des logs sous Unix et Linux. Il permet de définir la périodicité d'archivage des logs, mais aussi la durée pendant laquelle les archives seront conservées.

Dans certains cas, comme celui des systèmes militaires, la traçabilité des événements à long terme est très importante. Il est alors primordial de prévoir une sauvegarde efficace des journaux d'événements, si possible sous leur forme complète avant filtrage. Cela permet d'étudier finement ce qui s'est passé en cas de problème.

Fonction 4 : Filtrer, Analyser, Corréler les événements

Notre système de supervision doit traiter toute la masse d'information « brute » collectée pour qu'elle soit exploitable par l'administrateur. Pour cela, plusieurs approches sont possibles :

Dans la plupart des systèmes de détection d'intrusion comme Prelude, le filtrage et l'analyse se font en une seule passe : on recherche une liste d'événements connus pour être anormaux ou suspects, on les remonte comme alertes et on ignore le reste. Cette méthode a pour avantage d'être simple à mettre en œuvre, cependant elle repose entièrement sur la couverture de la liste des événements recherchés. Elle est adaptée à la détection d'intrusion, mais n'est pas satisfaisante pour une supervision plus globale, car on risque de ne pas voir des événements importants.

Pour les projets Log3C et SFS, on ne souhaite au contraire ignorer aucun événement inconnu. Cette approche consiste à filtrer d'abord tous les événements connus comme normaux et à remonter tout le reste. Ensuite, on poursuit l'analyse pour mettre en évidence des alertes pour des événements particuliers et il est également possible de corréler des événements liés pour faciliter la supervision.

Nous allons commencer par présenter les méthodes et outils employés pour l'analyse des événements, avant de préciser leur mise en œuvre.

(un peu de) regex

Pour reconnaître un événement et en extraire les informations intéressantes, la plupart des outils de supervision comme SEC, Prelude ou SFS emploient des expressions régulières, appelées encore « regex » ou « regexp ». Prenons par exemple un événement anodin :

```
09-11-01 11:17 INFO websrv: user toto successfully logged in.
```

Cet événement contient des parties fixes qui décrivent l'action, ici « `user` » et « `successfully logged in` », ainsi que des parties variables qui indiquent l'utilisateur concerné « `toto` », le serveur « `websrv` », le niveau de criticité « `INFO` », la date et l'heure de l'action.

Une expression régulière simple permettant de reconnaître cet événement serait :

```
INFO (\w+): user (\w+) successfully logged in.
```

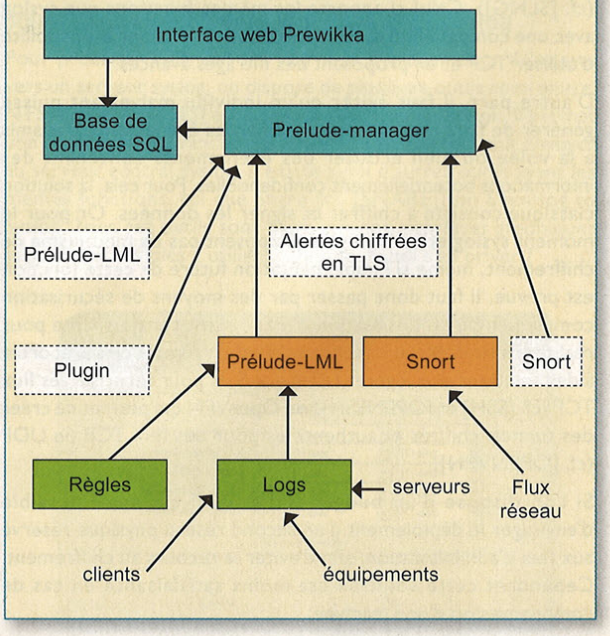
Lorsqu'elle est appliquée, cette expression régulière fournit en retour les variables `$1` et `$2`, qui correspondent respectivement au nom du serveur et au nom d'utilisateur extraits de l'événement.

Nous ne décrirons pas plus en détail le monde merveilleux et subtil des expressions régulières. Pour cela, rien ne vaut la lecture d'une bonne documentation comme le tutorial Perl [RETUT] et l'utilisation d'outils d'aide à la confection de regex, comme Regex-coach [RCOACH].

Prelude

Prelude est un système de détection d'intrusion open source (cf. [PRELUDE]). C'est un outil très intéressant pour la supervision de la sécurité car il fournit une infrastructure modulaire comprenant une grande partie des fonctionnalités nécessaires, de la collecte des événements jusqu'à l'affichage sur une console web de supervision.

Figure 3 *Architecture globale de Prelude*



Il permet notamment de centraliser des informations provenant de nombreuses sources comme les journaux d'évènements et de rechercher des motifs correspondant à des attaques connues dans ces informations.

Prelude-LML

Prelude peut recevoir les informations provenant d'une ou de plusieurs sondes. Parmi ces sondes, il existe « Prelude-LML », qui est un module permettant d'analyser les informations contenues dans des fichiers de log en temps réel. Dès qu'une ligne du fichier surveillé correspond à une règle basée sur une expression régulière, Prelude-LML extrait les informations utiles et transmet une alerte au format IDMEF (cf. [IDMEF]) au Prelude-Manager.

Prenons par exemple l'événement suivant, produit par un pare-feu Netscreen :

```
#system-emergency-00005: SYN flood! From 2.0.0.3:38254 to 20.0.0.3:74, proto TCP
(zone Untrust, int untrust). Occurred 1 times. (2002-01-31 00:01:51)
```

Voici une règle permettant au Prelude-LML de le détecter et d'en extraire les informations utiles :

```
regex=system-emergency-(\d+): (\S.+) From (\d.+):(\d+) to (\d.+):(\d+), proto
(\S.+)\(zone (\S+), int (\S.+)\). Occurred (\d+) times.; \
 classification.text=$2 from $3 to $5; \
 id=5000; \
 revision=1; \
 analyzer(0).name=Netscreen; \
 analyzer(0).manufacturer=Juniper; \
 analyzer(0).class=Firewall; \
 assessment.impact.severity=low; \
 assessment.impact.description=Emergency message $1:$2 from Netscreen
equipement; \
 source(0).node.address(0).category=ipv4-addr; \
 source(0).node.address(0).address=$3; \
 source(0).service.port=$4; \
 target(0).node.address(0).category=ipv4-addr; \
 target(0).node.address(0).address=$5; \
 target(0).service.port=$6; \
 last
```

Heureusement pour nous, de courageux volontaires ont déjà rédigé de nombreuses règles utiles pour superviser la plupart des logiciels et équipements standards. Le support de règles Netscreen a par exemple été ajouté suite à la contribution du projet Log3C, plus il y a de contributeurs et plus le projet s'améliorera !

Limitations de Prelude-LML

La principale limite de la version actuelle de Prelude-LML pour la supervision sécurité est qu'il n'existe pas de règle permettant de sélectionner *« tous les événements sauf ceux qui ont été reconnus »* (ceux qui ne correspondent à aucune des expressions régulières de la liste) et de les remonter comme alertes. Pour cela, le projet Log3C s'est appuyé sur SEC en complément de Prelude.

D'autre part, Prelude-LML est actuellement limité au traitement ligne par ligne des fichiers de log. C'est-à-dire qu'il réalise de la recherche d'information plus qu'il ne fait de la corrélation. Là aussi, SEC permet d'apporter des fonctions d'analyse complémentaires.

SEC

SEC, pour « *Simple Event Correlator* », est un outil d'analyse et de corrélation de fichiers log, écrit en Perl (cf. [SEC]). Son principe de fonctionnement est le suivant :

- On lui donne un fichier en entrée.
- Il vérifie si les données insérées en temps réel correspondent à des règles, en fonction de motifs écrits sous forme d'expressions régulières.
- Si c'est le cas, il effectue une action (écriture dans un autre fichier, lancement de script, e-mail…).

SEC utilise le même principe que Prelude-LML, les règles de SEC emploient des motifs recherchés dans les événements (expressions régulières, chaînes de caractères, fichier d'expressions régulières, négation d'une expression régulière…).

Mais la force de SEC vient du fait qu'il permet de corréler les événements entre eux. En effet, SEC dispose de plusieurs types de règles :

- **Single** – Règle qui exécute une action si un log correspond au pattern.
- **SingleWithScript** – Règle qui exécute une action si un log correspond au pattern et que le lancement d'un script extérieur ajoute une validation.
- **SingleWithSuppress** – Permet de générer une seule action pour x événements du même type en y secondes (fenêtre de validité).
- **Pair** – permet de combiner deux ou plusieurs patterns successifs.
- **PairWithWindow** – Permet de combiner des patterns successifs s'ils se produisent durant x secondes (fenêtre temporelle glissante).
- **SingleWithTreshold** – Permet de générer une seule action quand x ou plus d'événements se produisent durant y secondes (fenêtre temporelle glissante).
- **SingleWith2Tresholds** – Permet de générer une action quand x événements se produisent durant une fenêtre temporelle T1 puis à partir de cet instant ne pas générer d'action tant qu'il n'y a pas plus de y événements durant une fenêtre T2.
- **Suppress** – Supprime un événement.
- **Calendar** – Exécute une action à un instant spécifié.

Pour des traitements de corrélation plus « intelligents », il est même possible de lier plusieurs règles entre elles pour bâtir ce qui se rapproche d'un automate à états.

Exemples de règles avec corrélation :

Prenons un exemple simple. Sur un poste Windows standard, après 3 échecs de connexion sur un compte utilisateur, celui-ci est automatiquement verrouillé. Cependant le compte Administrateur local n'est jamais verrouillé pour éviter un blocage total de la machine, ce qui permet à un attaquant de tenter une recherche du mot de passe par force brute. Il serait donc intéressant de remonter une alerte particulière lorsque 3 échecs de connexion se produisent.

Avec NTsyslog, voici le type d'événement obtenu pour un échec de connexion sur un poste Windows en français :

```
May 12 17:05:12 2.0.0.4 security[failure] 529 AUTORITE NT\SYSTEM  Échec de
l'ouverture de session :    Raison : Nom d'utilisateur inconnu ou mot de
passe incorrect    Nom de l'utilisateur : Administrateur Domaine : TOKYO
Type de sessions : 2 Processus d'ouverture de session : User32 Package
d'authentification : Negotiate Nom de station de travail : TOKYO
```

La règle suivante est une règle SEC qui exécute une action si on détecte cet événement au moins 3 fois en moins de 4 minutes.

Elle écrit ensuite la ligne de log et le message « Multiple echec connexion administrateur » dans le fichier `/var/log/network/alertes.log` :

```
type=SingleWithThreshold
ptype=RegExp
pattern=Échec de l'ouverture de session :    Raison : Nom d'utilisateur inconnu
ou mot de passe incorrect    Nom de l'utilisateur : Administrateur
desc=Echec connexion Administreur
action=write /var/log/network/alertes.log $0 Multiple echec connexion
administrateur
window=240
thresh=3
```

Voici un autre exemple de règle qui corrèle 2 événements pour les remplacer par un 3ème de plus haut niveau :

```
type=Pair
ptype=RegExp
pattern=Le service d'Enregistrement d'événement a été arrêté.
desc=$0
action=logonly
ptype2=RegExp
pattern2=Le service d'Enregistrement d'événement a démarré.
desc2=$0
action2=write /var/log/network/alertes.log Redémarrage de Windows
window=120
```

Comme pour Prelude, il existe déjà de nombreuses règles écrites pour SEC, dont quelques exemples de corrélation plus complexes : cf. [SEC2].

Filtrer pour réduire la masse d'informations

Une fois que les événements sont centralisés, la première chose à faire est de réduire la masse d'information collectée en éliminant tous les événements non significatifs. En effet, toute machine génère des événements répétitifs, tout à fait normaux, qui alourdissent le travail du superviseur. Par exemple sous Windows, de nombreux services génèrent des événements lorsqu'ils démarrent ou qu'ils s'arrêtent. De même, une erreur de lecture sur un CD-ROM ne constitue pas une information pertinente pour la sécurité.

Dans SFS, les événements sont filtrés dès la source grâce à des expressions régulières, avant d'être transmis sur le réseau.

Dans Log3C, SEC est employé avec une première liste de règles permettant d'éliminer tous les événements reconnus comme non intéressants (règle « Suppress »).

Note : suivant la taille du système, il peut être utile de réduire les logs dès la source, ou bien sur des serveurs de centralisation intermédiaires, afin de soulager les serveurs finaux.

Analyser et Corréler

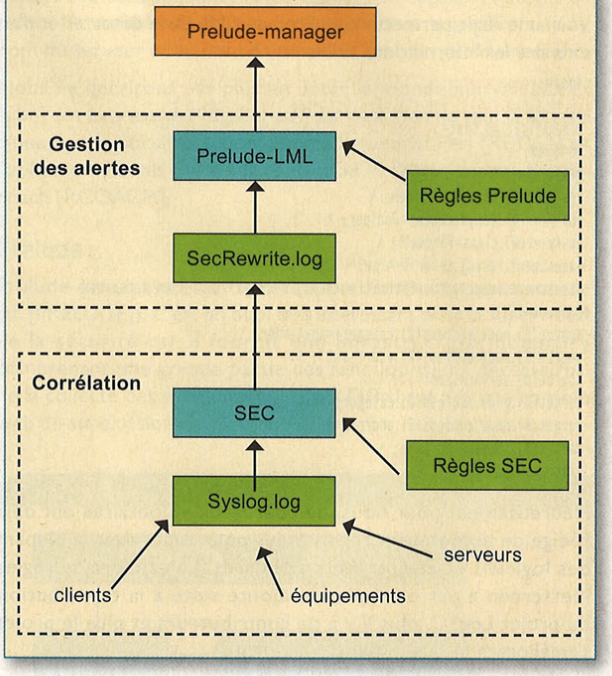
Nous avons indiqué qu'il était possible d'utiliser SEC en complément de Prelude-LML afin d'ajouter de la corrélation. Prenons un exemple simple où l'on se situe sur un serveur syslog.

Ce serveur centralise les informations dans un fichier nommé « Syslog.log ». Le fichier `Syslog.log` contient donc les logs de tous les équipements et systèmes du réseau. Nous appliquons ensuite des règles SEC sur le fichier `Syslog.log`.

Il en résulte un nouveau fichier `SecRewrite.log`. SEC permet de réécrire les événements, le fichier `SecRewrite.log` contient des événements concrets correspondant aux règles de corrélation (répétitions d'événements et scénarios d'attaque ne sont représentés que par une information unique).

Il suffit ensuite de configurer Prelude-LML pour traiter les données du fichier `SecRewrite.log` et formater les événements au format IDMEF (définir l'événement, son impact, sa criticité…) pour les envoyer au Prelude-Manager :

Figure 4 *Utilisation de SEC et Prelude-LML*



Cet exemple illustre donc le fonctionnement complémentaire de SEC et Prelude. C'est aussi cette méthode qui est utilisée dans le projet Log3C pour prendre en compte les événements inconnus. Il existe plusieurs façons de récupérer les événements inconnus, nous allons illustrer ceci dans la suite.

Nous allons, dans un premier exemple, supposer que l'administrateur met en place de la corrélation au niveau de SEC. SEC vérifie pour chaque nouvel événement si une de ses règles correspond à celui-ci. Dès qu'il trouve une règle lui correspondant, il réalise l'action décrite par la règle et passe à l'événement suivant (il ne vérifie pas les autres règles). Ceci permet de réaliser une règle finale correspondant à tous les événements restants (détectés par le motif « * »). Les événements correspondant à cette dernière règle étant supposés inconnus (puisque aucune règle ne leur correspond), il suffit d'écrire

les logs dans `SecRewrite.log` en ajoutant un identifiant de type `[unknown_log]`.

Ensuite il faut ajouter une règle au Prelude-LML pour qu'il formate correctement ces événements inconnus grâce à l'identifiant ajouté, par exemple :

```
regex=\[unknow_log\] \
 classification.text=événement inconnu; \
 id=5000; \
 revision=1; \
 […]
last
```

Imaginons un autre exemple où l'administrateur utilise uniquement les règles de Prelude-LML et non plus celles de SEC. Il peut utiliser SEC pour identifier les événements inconnus. Pour cela, il faut dans un premier temps réaliser un fichier `all.rules` contenant toutes les expressions régulières contenues dans les règles de Prelude-LML. Dans un second temps, il faut réaliser une règle SEC utilisant ce fichier pour « matcher » les événements connus, puis une règle finale correspondant à tous les événements restants (motif « * ») pour « matcher » les événements inconnus.

Voici à quoi peut correspondre un fichier de règles SEC à utiliser dans ce cas :

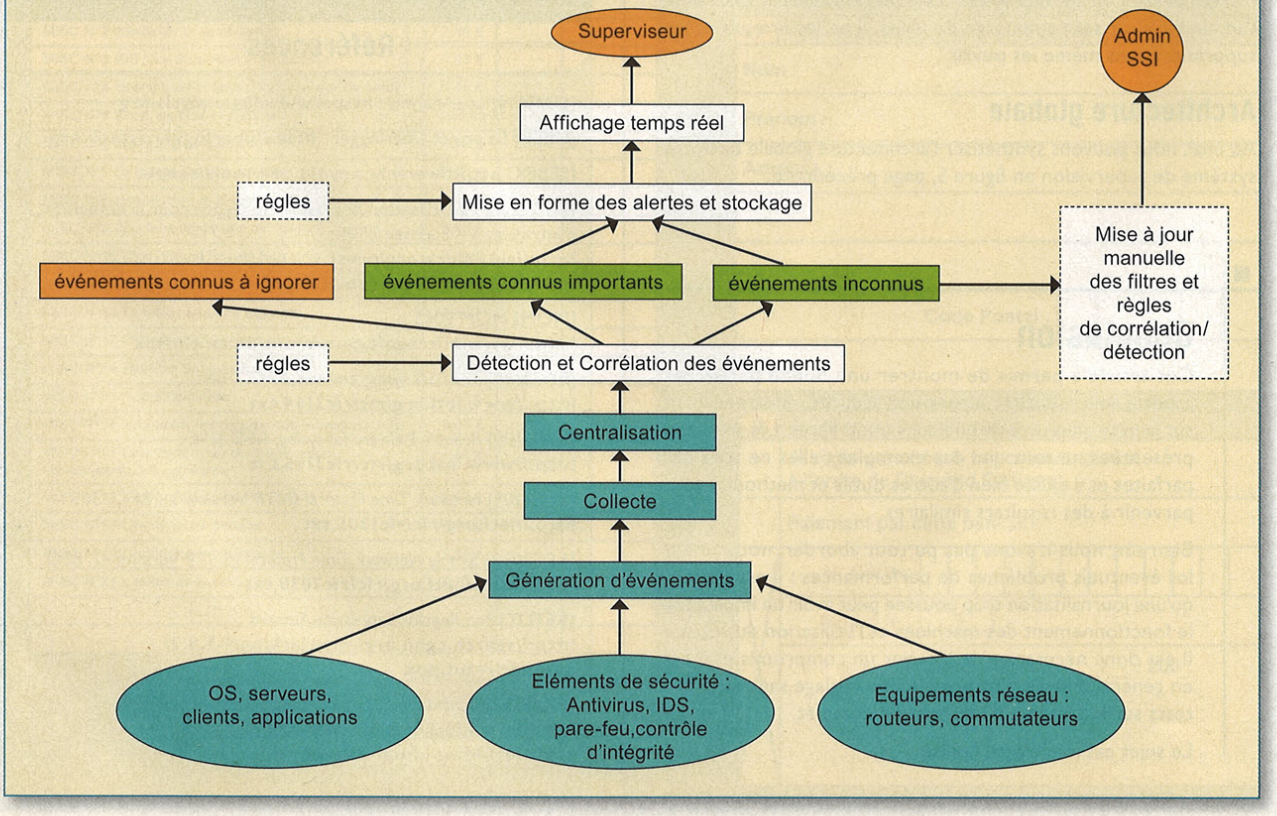
```
#Matching des événements connus
type=single
ptype=perlfunc
pattern=sub { my(@cat) = `cat all.rules`; chomp(@cat); \
              my($pat) = join('|', @cat); return ($_[0] =~ /($pat)/); }
desc=match patterns listed in a file
action=logonly
#Matching des autres événements:
type=single
ptype=perlfunc
pattern=sub { return $_[0]; }
desc=match everything
action=write inconnus.log $0 [unknown_log]
```

On obtient au final un fichier `inconnus.log` contenant tous les événements inconnus avec le tag `[unknow_log]`. Il suffit d'ajouter une règle au Prelude-LML pour prendre en compte tous les événements ajoutés au fichier `inconnus.log`, comme dans l'exemple précédent.

Fonction 5 : Afficher

Afficher correctement les alertes, c'est le but ultime de la corrélation, permettant ainsi à un administrateur d'avoir une vision réelle des événements importants se produisant sur son réseau. De nombreux outils fournissent des interfaces web pour afficher les données qu'ils gèrent. Dans le cas du projet Log3C, nous avons utilisé l'interface web « officielle » de Prelude. Cette interface se nomme « Prewikka » et récupère les informations dans la base de données SQL utilisée par le Prelude-manager.

Figure 5 *Architecture fonctionnelle du système de supervision*



Comme nous l'avons cité précédemment, Prelude-LML est limité au traitement ligne par ligne. On ne peut donc pas vraiment parler de corrélation (pour le moment). Par contre, l'interface web Prewikka permet de faire de l'agrégation automatique de données. L'agrégation permet de regrouper plusieurs événements ayant des caractéristiques communes. Par exemple, on peut regrouper les informations selon la machine sujette à une attaque ou encore par le type de sonde ayant généré l'alerte. Ceci offre une meilleure visibilité des alertes et simplifie donc la gestion des alertes. De même, les développements futurs de Prelude se porteront vers la corrélation d'événements. Pour le moment, la seule solution pour ajouter de la corrélation est d'utiliser un outil supplémentaire comme nous l'avons illustré.

Supervision par mail

Dans SFS, la machine qui centralise les évènements est la plus sensible. Stocker les évènements sur cette machine a deux inconvénients :

- Les administrateurs sont obligés de se connecter sur la machine pour visualiser les logs. Tout le monde sait pertinemment que cela durera un temps.
- Si la machine est compromise, les logs le seront aussi. Même si vos logs sont chiffrés, rien n'empêche le pirate de tout simplement les supprimer.

La solution retenue est alors d'utiliser le protocole SMTP pour exporter les logs. C'est facile à implémenter dans un script d'administration et tout le monde lit au moins une fois par jour ses mails. Par contre, si le traitement des évènements au départ a mal été pensé (filtrage des évènements peu efficace par exemple), l'administrateur sera submergé de mails, il ne les lira pas et les supprimera sans même les ouvrir.

Architecture globale

Au final, nous pouvons synthétiser l'architecture globale de notre système de supervision en figure 5, page précédente.

Conclusion

Cet article a permis de montrer une bonne partie des aspects concrets de la supervision sécurité, en se penchant sur la mise en œuvre de quelques outils libres. Les solutions présentées ne sont que des exemples : elles ne sont pas parfaites et il existe bien d'autres outils et méthodes pour parvenir à des résultats similaires.

Bien sûr, nous n'avons pas pu tout aborder, notamment les éventuels problèmes de performances : il est évident qu'une journalisation trop poussée peut avoir un impact sur le fonctionnement des machines et l'utilisation du réseau. Il est donc nécessaire de trouver un compromis et il est en général difficile d'estimer le bon réglage sans faire des tests sur le système d'information complet.

Le sujet est loin d'être épuisé… :-)

Boîte à outils

[ELFSH] Projet Elfsh: **http://elfsh.devhell.org**

[NTSL] NTsyslog : **http://ntsyslog.sourceforge.net/**

[SNARE] Snare : **http://www.intersectalliance.com/projects**

[LOGGER] Logger : **http://unixhelp.ed.ac.uk/CGI/man-cgi?logger+1**

[ADISCON] Adiscon : **http://www.monitorware.com/en/logger/index.php**

[SLNG] Syslog-NG : **http://www.balabit.com/products/syslog_ng/**

[SSH] SSH : **http://www.ssh.com**

[OPENSSH] OpenSSH : **http://www.openssh.com**

[OPENVPN] OpenVPN : **http://openvpn.net**

[KEYCHAIN] Keychain : **http://www.gentoo.org/proj/en/keychain/index.xml**

[RCOACH] Regex-coach : **http://www.weitz.de/regex-coach/**

[PRELUDE] Prelude IDS : **http://www.prelude-ids.org**

[SEC] SEC, Simple Event Correlator : **http://kodu.neti.ee/~risto/sec/**

[SEC2] Collection de règles pour SEC : **http://www.bleedingsnort.com/sec/**

Références

[LOGAN] LogAnalysis : **http://www.loganalysis.org**

[UBER] Le projet Uberlogger : **http://uberlogger.rstack.org**

[SEBEK] **http://www.honeynet.org/tools/sebek**

[EUROSEC] « Utilisation des outils Honeypot pour la détection d'intrusion » – Eurosec 2005 : **http://sid.rstack.org/pres/0503_Eurosec_HoneypotIDS.pdf**

[MOM] MOM 2005 : **http://www.microsoft.com/mom/default.mspx**

[RFC3164] *The BSD syslog protocol*, RFC 3164 : **http://www.ietf.org/rfc/rfc3164.txt**

[RFC3195] *Reliable Delivery for syslog*, RFC 3195 : **http://www.ietf.org/rfc/rfc3195.txt**

[RFC1305] *Network Time Protocol* (NTP Version 3), RFC 1305 : **http://ietf.org/rfc/rfc1305.txt**

[RFC2030] *Simple Network Time Protocol* (SNTP Version 4), RFC 2030 : **http://ietf.org/rfc/rfc2030.txt**

[RETUT] *Perl Regular Expression Tutorial*, **http://search.cpan.org/~nwclark/perl-5.8.2/pod/perlretut.pod**

[IDMEF] *The Intrusion Detection Message Exchange Format* : **http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-14.txt**

Offre Collectionneur !

4 façons de commander :

- par courrier postal en nous renvoyant le bon ci-dessous
- par le Web, sur **www.ed-diamond.com**
- par téléphone, entre 9h-12h et 14h-17h au 03 88 58 02 08
- par fax au 03 88 58 02 09 (CB)

Vous êtes un fidèle lecteur mais vous ne vous rappelez plus dans quel magazine vous avez lu un article sur ... ?

Un sujet vous passionne et vous recherchez des magazines traitant de ce sujet ?

Allez sur **www.ed-diamond.com** et utilisez le moteur de recherche sur tous les sommaires des magazines édités par Diamond Editions (Misc, Linux Magazine et hors série, Linux Pratique). Vous pourrez également compléter votre collection !

Bon de commande à remplir et à retourner à :

Diamond Editions - Service des Abonnements/Commandes 6, rue de la Scheer B.P. 121 - 67603 Sélestat Cedex

DÉSIGNATION	PRIX	QTÉ	TOTAL
MISC N°1 Les vulnérabilités du Web !	5,95 €		
MISC N°2 Windows et la sécurité	7,45 €		
MISC N°3 IDS : La detection d'intrusions	7,45 €		
MISC N°4 Internet, un château construit sur du sable	7,45 €		
MISC N°5 Virus, mythes et réalités	Epuisé		
MISC N°6 Insécurité du wireless?	7,45 €		
MISC N°7 La guerre de l'information	7,45 €		
MISC N°8 Honeypots ; le piège à pirates	7,45 €		
MISC N°9 Que faire après une intrusion ?	7,45 €		
MISC N°10 VPN (Virtual Private Network)	7,45 €		
MISC N°11 Tests d'intrusion	7,45 €		
MISC N°12 La faille venait du logiciel !	7,45 €		
MISC N°13 PKI - Public Key Infrastructure	7,45 €		
MISC N°14 Reverse Engineering	7,45 €		
MISC N°15 Authentification	7,45 €		
MISC N°16 Télécoms, les risques des infrastructures	7,45 €		
MISC N°17 Comment lutter contre le spam, les malwares, les spywares	7,45 €		
MISC N°18 Dissimulation d'informations	7,45 €		
MISC N°19 Les dénis de service	7,45 €		
MISC N°20 Crytographie malicieuse	7,45 €		
MISC N°21 Limites de la sécurité	7,45 €		
		TOTAL	
		Frais de port France Metro : + 3,81 €	
		Frais de port Etranger : + 5,34 €	
		TOTAL	

Oui je souhaite compléter ma collection

1 Voici mes coordonnées postales

Nom :

Prénom :

Adresse :

Code Postal :

Ville :

2 Je joins mon réglement :

Je règle par chèque bancaire ou postal à l'ordre de Diamond Editions*

Paiement par carte bancaire :

N° Carte :

Expire le : __/__/

Date et signature obligatoire : __/__/ 200_

La gestion des correctifs de sécurité

La gestion des correctifs de sécurité, ou patch management, est aujourd'hui l'un des éléments clés de la protection des Systèmes d'Information. Il y a quelques années encore, la gestion des patches n'était nécessaire que sur un sous-ensemble relativement restreint de systèmes : principalement les serveurs les plus sensibles et/ou exposés, comme les serveurs accessibles depuis Internet par exemple. Malheureusement, aujourd'hui, ce n'est plus aussi simple.

La cause principale en est la diffusion récurrente de virus et autres vers auto-propagateurs. La menace a en effet évolué avec l'apparition de vers exploitant des failles de sécurité de logiciels largement répandus sur le marché, comme CodeRed, Nimda, Blaster, ZoTob, etc. Le périmètre vulnérable s'est par la même occasion considérablement élargi : il ne suffit plus de protéger les serveurs sensibles ou exposés, car toute machine (par exemple le PC bureautique d'un utilisateur) est une cible potentielle et peut se transformer à son tour en source de contamination. L'impact unitaire n'est, il est vrai, que de faible intensité, mais lorsque des milliers de PC sont infectés par des vers balayant des plages d'adresses entières à la recherche de machines vulnérables, c'est tout le réseau interne de l'entreprise qui peut être mis hors service pendant plusieurs jours, sans parler des éventuels effets de bord des vers ou virus en question (destruction de fichiers, diffusion à l'extérieur de documents confidentiels situés dans le répertoire « Mes Documents » ou sur le Bureau, etc.).

L'étude des principaux vers ayant affecté les entreprises du monde entier durant ces dernières années montre par ailleurs que la situation ne fait qu'empirer. En effet, de CodeRed à ZoTob par exemple, la période de gestation des vers (entre la publication de la vulnérabilité par l'éditeur concerné et l'apparition d'un ver exploitant la faille en question) s'est considérablement raccourcie : 11 mois pour Nimda, 6 mois pour SQL Slammer et seulement trois semaines pour Blaster. On peut même citer le cas du ver Witty, affectant les produits ISS, apparu dans la nature le lendemain de la publication officielle de la vulnérabilité qu'il exploitait !

De plus, la vitesse de propagation des vers (entre l'apparition des premières instances du ver et l'atteinte du nombre maximal de machines contaminées dans le monde) s'accélère : quelques jours pour CodeRed, quelques minutes seulement pour SQL Slammer.

La gestion des correctifs de sécurité

Ainsi, il est nécessaire d'**agir** si l'on ne veut pas perdre son temps (et son argent) à **réagir** en cas de contamination : il s'agit d'être proactif plutôt que réactif. Cela n'empêche pas, bien sûr, de prévoir des processus d'urgence en réponse à la sortie d'un nouveau ver exploitant une vulnérabilité répandue ou d'un nouveau virus, d'où l'intérêt d'intégrer le processus de *patch management* dans son SOC (*Security Operations Center*) et dans sa supervision de la sécurité en général (voir les autres articles sur la supervision de la sécurité dans ce dossier).

La problématique de la gestion des correctifs de sécurité semble évidente au premier abord : il « suffit » d'installer régulièrement les patches diffusés par les éditeurs pour corriger les vulnérabilités de leurs systèmes d'exploitation, produits et applications. Cependant, dans la pratique, les entreprises se heurtent rapidement à de nombreuses difficultés et celles-ci croissent de manière proportionnelle à la taille et à la complexité des Systèmes d'Information de l'entreprise : temps de réaction trop longs, multiplication des types de systèmes et des vulnérabilités associées, manque d'expertise technique, problèmes de régression, coûts de déploiement, gestion des matériels nomades, etc. Il est donc nécessaire d'industrialiser le processus de gestion des correctifs dans le contexte de son entreprise.

Une solution consiste à définir des stratégies de gestion proactives des correctifs, adaptées aux menaces et aux contraintes métier de l'entreprise, et à les implémenter à l'aide d'outils automatisés disponibles sur le marché. Nous allons donc, dans cet article, commencer par décrire différents types de stratégies de gestion des correctifs de sécurité, en se plaçant dans l'hypothèse d'une grande entreprise, puis nous ferons un panorama rapide des outils de gestion des correctifs, en examinant plus en détail les outils fournis par Microsoft.

Les stratégies de gestion des correctifs de sécurité

La problématique de la gestion des correctifs de sécurité dépend du contexte dans lequel on se place. En effet, la stratégie de patch management sera différente pour la production informatique et pour les postes bureautiques par exemple. De même, la stratégie d'application récurrente des patches sera différente de la gestion de crise utilisée en cas d'apparition d'une vulnérabilité critique ou d'un ver sur le réseau interne.

Dans les paragraphes suivants, nous allons commencer par analyser le processus habituel, récurrent, de la gestion des correctifs, en explicitant les différences induites par divers contextes d'exploitation.

Le processus récurrent

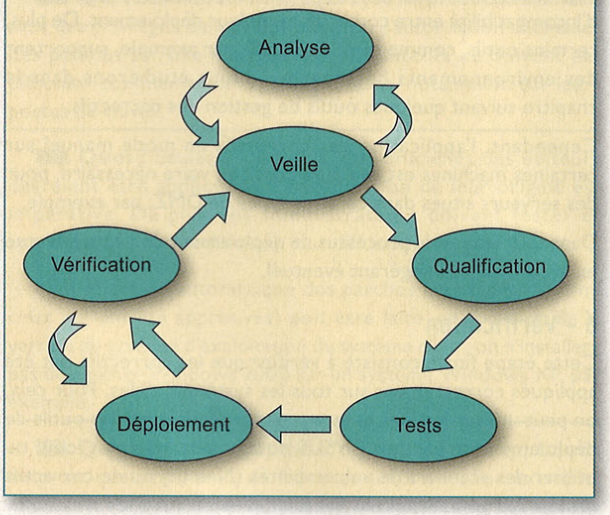
Nous nous plaçons ici dans un mode **proactif**. En effet, la plupart des éditeurs diffusent leurs correctifs à intervalle régulier, afin de permettre aux administrateurs d'effectuer des cycles de tests et de diffusion des patches, afin que les correctifs individuels soient diffusés ensemble.

Patrick CHAMBET
Architecte Sécurité des Systèmes d'Information, Bouygues Telecom
patrick@chambet.com
http://www.chambet.com

Microsoft diffuse par exemple ses correctifs de sécurité « non urgents » une fois par mois (le deuxième mardi de chaque mois) et Oracle publie ses correctifs majeurs tous les 3 mois (« *Oracle Critical Patch Update* »).

Le processus de gestion des correctifs est un processus itératif. Les étapes suivantes doivent en particulier être respectées :

Figure 1 *Processus de gestion des correctifs*



1 - Analyse

Il est nécessaire avant toute chose d'analyser l'existant dans l'entreprise, dans les différents types d'environnements (production, pré-production, intégration, bureautique, réseau...) et d'effectuer une étude de risques mettant notamment en valeur les processus et les systèmes les plus critiques.

Un inventaire exhaustif du parc, y compris les ordinateurs portables (source majeure de problèmes dans bien des cas) permettra d'avoir toutes les informations nécessaires sur les types de systèmes et d'applicatifs, leur version, leur emplacement (DMZ ou cœur de production par exemple), etc. Un outil d'inventaire et de gestion de parc est le plus souvent indispensable pour recueillir ces informations et, surtout, pour maintenir leur cohérence dans le temps.

Il est également possible d'utiliser des scanners de vulnérabilités de type Nessus, bien qu'on commence à empiéter ici sur les étapes suivantes (voir article sur les VDS dans ce même numéro).

2 - Veille sécurité

Une fois l'inventaire matériel et logiciel établi, l'objectif de la veille sécurité est d'identifier les nouvelles vulnérabilités découvertes et/ou publiées concernant l'environnement existant, puis de trouver les mises à jour éventuelles mises à disposition par les éditeurs ou les fournisseurs, et ce, de manière fiable.

Cette veille sécurité peut être effectuée par l'équipe sécurité interne, par l'entité chargée de la gestion du parc informatique ou encore être externalisée.

3 - Qualification

L'objectif de la qualification est, d'une part, de déterminer l'exploitabilité des vulnérabilités identifiées à l'étape précédente et leurs impacts potentiels, et, d'autre part, d'évaluer la pertinence de la diffusion des correctifs dans les différents environnements de l'entreprise et de définir si le processus à utiliser est le processus normal ou urgent.

Cette étape est donc primordiale : c'est à ce stade qu'il va être décidé si un correctif va être diffusé dans l'entreprise ou non, et si oui, dans quelles conditions et sur quel périmètre. En effet, le contexte du déploiement futur va influer à la fois sur l'évaluation de la criticité de la vulnérabilité et sur le coût de déploiement du correctif.

Prenons comme exemples les contextes de la bureautique et de la production.

Contexte de la bureautique

Le contexte de la bureautique est, d'une certaine manière, plus simple. En effet, les correctifs publiés par Microsoft sont à l'heure actuelle testés de manière extrêmement approfondie, et les effets de bord, lors de leur déploiement, sont de plus en plus rares. Un grand nombre d'entreprises a fait le choix d'appliquer les correctifs de sécurité diffusés par Microsoft (et surtout ceux qualifiés de critiques) sur l'ensemble de son parc Windows après un cycle de tests réduit, surtout en ce qui concerne les postes de travail. Dans la pratique, les problèmes de compatibilité avec les applications clientes installées sur les postes de travail sont très rares, et sont, le plus souvent, le fait de versions très anciennes de progiciels. En cas d'apparition d'un ver sur le réseau interne, tentant d'exploiter les vulnérabilités déjà *patchées*, le gain en fiabilité est évident.

Contexte de la production informatique

Dans un contexte de production, par contre, la qualification des correctifs doit être plus prudente et plus élaborée. Il n'est plus possible de déployer des correctifs sur des serveurs applicatifs sans s'assurer au préalable que les impacts sur leur fonctionnement sont compatibles avec les contrats de service.

De plus, sur des parcs de très grande taille (plusieurs milliers de serveurs), il n'est souvent pas possible d'appliquer tous les correctifs diffusés par les éditeurs, pour deux raisons majeures :

- Appliquer tous les correctifs nécessiterait des arrêts et redémarrages permanents des systèmes en production.

→ Le coût du parc augmenterait de façon importante, sans apporter de gains suffisamment substantiels sur la fiabilité.

A cela s'ajoute la problématique des systèmes obsolètes, pour lesquels les éditeurs ne diffusent plus de correctifs de sécurité (Windows NT 4.0 par exemple). Cela peut survenir lors de la reprise de l'existant sur un parc de grande taille dont l'historique est conséquent, ou bien lorsque le parc contient des équipements de type « boîtes noires », contenant des éléments logiciels dont on ne maîtrise pas la nature (équipements Ericsson, par exemple). De plus, dans ce dernier cas, il est même interdit de mettre à jour les systèmes tournant dans ce type de solution packagée, sous peine de perdre la garantie de l'éditeur !

Pendant l'étape de qualification, qui est faite en général par l'équipe sécurité ou le bureau d'études interne, l'implication des maîtrises d'œuvre des applications impactées est forte dans l'environnement de production : en effet, ce sont elles qui possèdent la connaissance des applications qui tournent sur les serveurs, et qui pourront donc qualifier les impacts applicatifs éventuels du correctif. C'est surtout le cas pour les progiciels (SAP, Siebel,...), un peu moins dans le cas de correctifs des OS sous-jacents.

Dans le cas d'Oracle, les impacts fonctionnels des patches peuvent être importants : en particulier, les correctifs Oracle, assez volumineux, requièrent souvent la mise en adéquation d'un grand nombre de dépendances. Ainsi, une mise à jour s'apparente plus à un projet de migration qu'à une simple mise à jour de sécurité. La validation par la maîtrise d'œuvre des applications est obligatoire dans ce cas.

4 - Tests

Une fois la qualification effectuée, il convient de tester l'application du correctif de sécurité sur un environnement suffisamment représentatif du périmètre cible. Pour les postes de travail, quelques postes installés avec les masters de l'entreprise, contenant les principales applications utilisées par les utilisateurs, seront suffisants.

Pour la production, on pourra bien sûr utiliser les environnements de tests applicatifs. Ensuite, une période d'observation sur un environnement de pré-production permettra de s'assurer de l'absence d'effets de bord ou de problèmes de régression.

Il ne faut pas oublier non plus de tester la désinstallation du correctif, au cas où un retour en arrière serait nécessaire plus tard.

5 - Planification et déploiement

C'est à cette étape que se préparent et s'exécutent le déploiement et l'application des correctifs sur les systèmes cibles.

Les points à prendre en compte lors de la planification sont notamment les suivants :

- Programmer les téléchargements par zones géographiques, afin d'éviter la saturation du réseau lors des transferts de correctifs. Dans tous les cas, un contrôle de la bande passante doit être effectué.
- Certains correctifs doivent être appliqués dans un ordre précis.
- Tenir compte des éventuels redémarrages après application des patches : en effet, certains correctifs nécessitent le *reboot* des machines après leur application.
- Une politique spécifique doit être mise en place pour les postes nomades (de l'exclusion du réseau des postes non à jour au « patchage » automatique).
- Les créneaux de maintenance prévus sur les différentes cibles doivent être utilisés en priorité.

Sur un parc de grande taille, il est recommandé de s'appuyer sur un logiciel de déploiement durant cette étape, afin d'éviter que celle-ci s'étale sur une trop longue période, même en mode proactif. Les bons logiciels de gestion des correctifs comportent une fonctionnalité de hiérarchisation des correctifs selon le degré de criticité des vulnérabilités, et détectent les problèmes d'incompatibilité entre correctifs avant leur déploiement. De plus, certains outils, comme ceux de Shavlik par exemple, supportent des environnements hétérogènes. Nous étudierons dans le chapitre suivant quelques outils de gestion des correctifs.

Cependant, l'application des correctifs en mode manuel sur certaines machines est toujours possible, voire nécessaire, pour des serveurs situés dans certains types de DMZ, par exemple.

Dans tous les cas, le processus de déploiement doit être effectué en accord avec l'infogérant éventuel.

6 - Vérification

Cette étape finale consiste à vérifier que les correctifs ont été appliqués correctement sur tous les systèmes cibles. Pour cela, on peut utiliser les logs et les rapports générés par les outils de déploiement de patches (WSUS notamment, voir plus loin), ou utiliser des scanners de vulnérabilités (Nessus) ou de correctifs (MBSA 2.0 pour Windows par exemple).

Si des systèmes sur lesquels l'application des correctifs a échoué sont détectés, il convient alors d'en analyser la raison et de tenter un redéploiement de ces correctifs.

Cette vérification doit être effectuée régulièrement, afin de détecter l'éventuelle apparition de systèmes non à jour, par exemple.

Processus urgent

Le mode de déploiement urgent d'un correctif est un mode réactif utilisé lorsque, à l'étape de veille, un risque important est identifié. Il peut s'agir de la diffusion d'un ver sur le réseau interne, à partir du portable d'un prestataire par exemple (bien que la connexion d'un tel type de poste de travail doive être interdite). Cela peut aussi être le cas lorsque des serveurs accessibles depuis Internet sont vulnérables (exemple : vulnérabilité Apache, IIS ou encore une vulnérabilité de la pile TCP/IP, comme la vulnérabilité TCP CAN-2004-0230).

Dans ce cas, l'ensemble des étapes suivantes (qualification et tests en urgence, déploiement automatique ou manuel immédiat) doit être effectué en moins de 48h. De plus, dans le cas d'un ver, des mesures complémentaires de filtrage peuvent être prises, au niveau des *firewalls*, routeurs, firewalls personnels, etc.

Les outils de gestion des correctifs de sécurité

Nous avons vu que les approches classiques, basées sur des actions manuelles (`apt-get`) ou semi-automatiques (Microsoft Update, RHN) ne sont pas suffisantes à l'échelle d'une grande entreprise. La gestion des patches doit donc être automatisée de façon à faciliter le suivi de leur application, à être réactif en cas d'apparition d'une nouvelle menace (nouvelle attaque par exemple), à les tester avant leur déploiement et à faciliter le retour en arrière en cas de problème.

L'utilisation d'un outil de gestion des correctifs doit respecter un certain nombre de conditions, et notamment : Qui ? Quoi ? Quand ? Où ? Comment ?

Qui ? Seuls les administrateurs (ou les processus tournant avec des privilèges élevés) doivent avoir l'autorisation d'installer des patches sur une machine. Les utilisateurs ne doivent pas effectuer eux-mêmes ce type d'opération, notamment sur leurs postes de travail.

Quoi ? Seules les mises à jour officielles des éditeurs devraient être appliquées. La vérification de leur origine est impérative. De plus, les administrateurs doivent tester et approuver les correctifs avant leur diffusion.

Une sélection automatique des patches nécessaires (parmi ceux qui ont été approuvés) doit être faite, en fonction de la version du système d'exploitation du système cible : on n'installera pas de patches Windows 2000 sur un système Windows XP, par exemple.

Quand ? Les patches doivent être appliqués automatiquement, mais de façon coordonnée. Par exemple, Microsoft recommande d'appliquer les patches critiques sous 24 heures, ceux classés « importants » sous un mois, ceux concernant une faille d'importance « modérée » sous 4 mois et les autres (faible importance) sous un an.

Où ? Sur les périmètres qualifiés. Même les PC portables doivent être patchés. C'est la raison pour laquelle il est impératif qu'ils se reconnectent régulièrement au réseau de l'entreprise.

D'où les correctifs doivent-ils être téléchargés ? Depuis un ou plusieurs serveurs centraux, en utilisant typiquement le plus proche.

Comment ? Les patches sont appliqués automatiquement, en tâche de fond, de façon à ce que l'opération soit transparente pour l'utilisateur. Si nécessaire, un redémarrage du système devra être planifié. Des informations pertinentes doivent être enregistrées dans des logs pendant et après l'opération.

Certains outils supportent en standard des plateformes hétérogènes : PatchLink, par exemple, supporte Windows, Unix (Solaris, IBM AIX, et HP UX), Linux, Macintosh et NetWare. Certains nécessitent le déploiement d'un agent sur les machines, d'autres non. Les agents offrent souvent une plus grande richesse fonctionnelle (SMS, par exemple) et consomment moins de bande passante, mais le coût de leur déploiement doit être évalué.

Environnements Microsoft

Microsoft propose plusieurs outils de gestion des correctifs :

- Scan des patches sur les machines : MBSA 2.0 ;
- Mise à jour manuelle et semi-automatique : Microsoft Update ;
- Mise à jour automatique : WSUS (*Windows Server Update Services*) et SMS (*Systems Management Server*).

Nous allons étudier plus en détail ces outils dans les paragraphes qui suivent.

MBSA

MBSA 2.0 (*Microsoft Baseline Security Analyzer*) est l'un des outils Microsoft les plus simples. Il est issu à l'origine de la société Shavlik. Une version gratuite de cet outil peut être téléchargée à l'URL suivante : **http://www.microsoft.com/technet/security/tools/mbsahome.mspx**

MBSA 2.0 peut effectuer des scans en local ou à distance sur des systèmes Windows 2000 SP3 et plus, Windows XP et Windows Server 2003. En plus des patches de sécurité Windows, il supporte également un grand nombre de produits Microsoft :

- Microsoft Office XP et plus ;
- Exchange Server 2000 et plus ;
- SQL Server 2000 et plus.

Et les autres produits supportés par Microsoft Update (voir : **http://support.microsoft.com/?scid=kb;en-us;895660**)

MBSA peut être exécuté en mode graphique (lancer `mbsa.exe`) ou en ligne de commande (lancer `mbsacli.exe`). Dans le second mode, il est possible d'utiliser des fichiers *batches* afin d'automatiser l'outil.

Par exemple, le script qui suit scanne un système et enregistre les résultats dans un fichier XML :

```
set cname=%computername%
set uname=%username%
"C:\Program Files\MBSA\mbsacli.exe" /nvc /nosum /c %cname% /n
  IIS+OS+SQL+Password /o %cname%
copy "%userprofile%\SecurityScans\%cname%.xml"
  "\\%cname%\c$\Documents and Settings\%uname% \SecurityScans\"
```

Fonctionnement de MBSA

Voici les différentes étapes du processus de vérification de MBSA lorsqu'il est lancé :

1. MBSA analyse la configuration de sécurité du système analysé. Il détecte les erreurs de configuration les plus fréquentes telles que :
   - les partitions en FAT ;
   - les comptes Administrateurs ;
   - les mots de passe triviaux ;
   - les services activés qui peuvent être dangereux ;
   - les partages de fichiers ;
   - la politique d'audit ;

- la configuration du firewall personnel (en local uniquement) ;
- etc.

Pour une liste complète des tests de sécurité effectués par MBSA, se reporter au fichier `Checks.csv` situé dans le répertoire de MBSA.

2 MBSA télécharge ensuite une référence de sécurité au format XML : il s'agit en fait d'un fichier nommé « mssecure.xml ». MBSA peut télécharger ce fichier directement depuis Internet ou à partir d'un serveur WSUS interne.

Depuis Internet, MBSA essaye successivement les liens suivants :

- **http://go.microsoft.com/fwlink/?LinkId=18922**
- **http://download.microsoft.com/download/xml/security/1.0/nt5/en-us/mssecure.cab** (version 3.32)

Le fichier CAB contient une version compressée du fichier `mssecure.xml`.

Notez qu'il est également possible de télécharger le dernier référentiel de sécurité à partir des liens suivants sur le site de shavlik :

- **http://xml.shavlik.com/mssecure.cab** (version 4.0)
- **http://xml.shavlik.com/mssecure.xml**

Si MBSA ne peut pas télécharger le fichier `mssecure.xml`, il utilise la copie locale (la dernière version téléchargée en local). Ainsi, vous pouvez télécharger le fichier `mssecure.xml` de Shavlik et l'utiliser avec MBSA. Mais notez que c'est une opération non supportée par Microsoft.

3 Puis MBSA analyse le niveau de patches du système scanné par rapport au référentiel de sécurité.

4 MBSA détecte les patches de sécurité manquants, ainsi que les Service Packs et affiche les messages correspondants dans son rapport.

Le fichier `mssecure.xml` est un fichier précieux : il contient tous les correctifs de sécurité publiés depuis 1998, avec des informations descriptives. Pour chaque patch, sont indiquées notamment les informations suivantes :

- description ;
- chemin d'accès au fichier de mise à jour ;
- chemin, version et somme de contrôle du patch ;
- les clés de registre modifiées par le patch.

Le fichier `mssecure.xml` contient aussi un historique des anciens correctifs inclus depuis dans des patches cumulatifs ou des Service Packs. Ce fichier XML est bien sûr modifié à chaque fois qu'un nouveau correctif de sécurité est publié.

Scripter MBSA

Les scans de MBSA peuvent être automatisés en utilisant des scripts : vous pouvez ainsi réaliser des tests à grande échelle.

Pour plus d'information, voir :

http://www.microsoft.com/technet/security/tools/mbsascript.mspx

Vous pouvez par exemple télécharger les scripts `batchscan.js` et `rollup.js`, qui permettent de scanner un nombre illimité de systèmes ou d'adresses IP à partir d'un fichier, tout en compilant les résultats dans un rapport de synthèse unique (fichier XML) qui peut être visualisé à l'aide d'Internet Explorer.

Microsoft Update

Microsoft Update est un outil de vérification et d'installation en ligne de patches qui peut être utilisé selon deux modes : un mode manuel, avec Internet Explorer pointant sur l'adresse **http://update.microsoft.com/microsoftupdate/** et un mode automatique ou semi-automatique, utilisant le service « *Automatic updates* » comme partie cliente. Microsoft Update est idéal pour les entreprises de petite taille.

Lorsqu'il est utilisé en mode automatique, le service « Automatic Updates », qui permet une mise à jour automatique et en tâche de fond, nécessite que le service BITS (*Background Intelligent Transfer Service*) soit activé : ce service utilise la bande passante non exploitée afin de télécharger les patches depuis le site web de Microsoft, de façon à rendre le processus totalement transparent pour l'utilisateur.

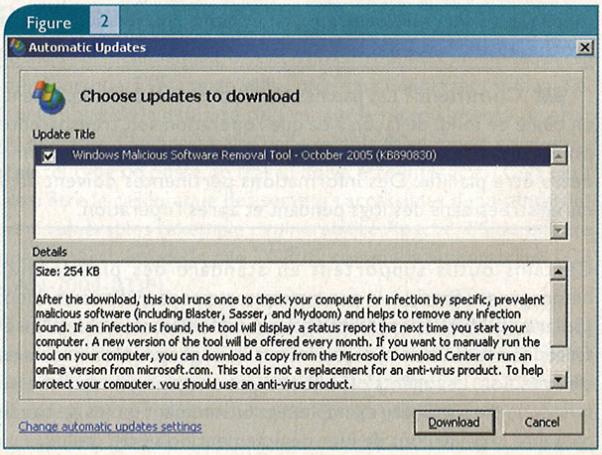
Le client Microsoft Update vérifie que chaque patch a été correctement installé sur l'ordinateur local. Pour ce faire, il effectue des vérifications à plusieurs niveaux :

- clés de registre (situées dans `HKLM\SOFTWARE\Microsoft\Updates\Windows` [VERSION]\SP[X]\KBxxxxxx) ;
- liste de fichiers sur le disque ;
- version et somme de contrôle de ces fichiers.

Le processus de mise à jour tourne en tâche de fond et reste transparent pour un utilisateur normal. Les notifications de nouveaux patches sont présentées à l'utilisateur logué localement seulement s'il a les privilèges Administrateur :

Ce type de notification a exactement le même aspect qu'avec le client WSUS (voir ci-après).

Figure 2



Windows Server Update Services

WSUS est un outil gratuit qui peut être téléchargé à l'URL suivante :

http://www.microsoft.com/windowsserversystem/updateservices/downloads/WSUS.mspx

Contrairement à MSUS 1.0, qui ne gérait que les correctifs Windows, WSUS peut maintenant gérer les correctifs des produits suivants :

- Windows 2000 SP3+, XP et Server 2003 ;
- Office (XP SP2 et 2003) ;
- SQL Server 2000 ;
- Exchange Server 2003.

Le principe de WSUS est d'avoir un serveur « Microsoft Update dans votre entreprise » : un ou plusieurs serveurs internes hébergent les patches de sécurité. A chaque fois que de nouveaux correctifs de sécurité sont publiés, l'administrateur approuve les patches nécessaires. Ceux-ci sont ensuite téléchargés sur les serveurs WSUS internes. Puis les postes utilisateurs se connectent automatiquement à l'un des serveurs internes de façon à télécharger et appliquer les patches validés.

WSUS utilise une interface d'administration web (**http://wsus.votre-intranet.com/WSUSAdmin/**) et a besoin d'IIS 6.0 sur les serveurs WSUS Windows Server 2003 internes.

Fonctionnement de WSUS

WSUS fonctionne un peu comme MBSA, même si leurs formats sont différents : WSUS a aussi besoin d'un référentiel de sécurité. Chaque jour, WSUS effectue un processus de synchronisation, en suivant les étapes ci-dessous :

- WSUS télécharge un référentiel de sécurité (fichiers XML).
- Il valide la signature de Microsoft des CAB.
- Il compare ce référentiel au contenu de sa base locale de façon à identifier les nouvelles mises à jour.
- Il attend l'approbation de l'administrateur.
- Il télécharge les patches approuvés par l'administrateur (uniquement) et vérifie leur signature.
- Il met à jour ses journaux de synchronisation et d'approbation.

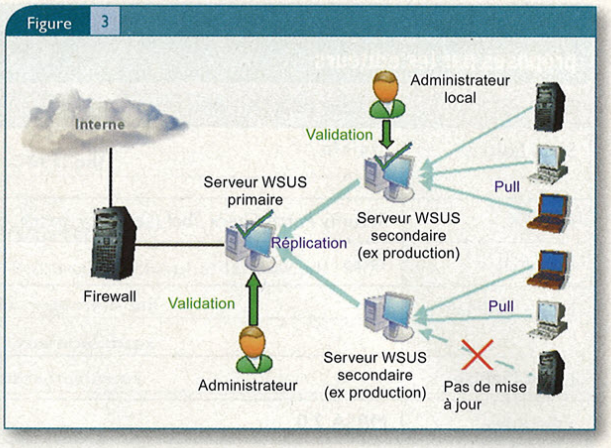
Si la synchronisation programmée échoue, WSUS réessaye trois fois avec à 30 minutes d'intervalle.

Une architecture WSUS avancée ressemble à la figure 3.

WSUS est un outil très puissant qui permet d'implémenter la politique typique que nous avons vue précédemment en répondant aux différentes questions posées :

Qui ? La partie cliente, le service « Automatic Updates », tourne avec les privilèges SYSTEM. Si un utilisateur est Administrateur, il a le choix d'appliquer ou pas les patches (voir plus haut). Les utilisateurs normaux ne peuvent pas refuser l'application automatique des patches, mais ne peuvent pas installer des patches de leur propre initiative.

Figure 3



D'ailleurs, une fois que WSUS est installé sur votre réseau interne, il est recommandé d'interdire les domaines correspondant à Microsoft Update au niveau du proxy sortant afin qu'aucun utilisateur ne puisse mettre à jour une machine de sa propre initiative. Ces domaines sont notamment les suivants :

- **http://www.windowsupdate.com**
- **http://windowsupdate.microsoft.com**
- **http://update.microsoft.com**

Quoi ? Le serveur WSUS vérifie la signature des mises à jour afin d'être sûr qu'ils ont bien été émis par Microsoft. Les mises à jour appropriées sont alors automatiquement sélectionnées (parmi celles qui ont été validées par l'administrateur), en fonction de l'ordinateur cible (version de l'OS, langue, etc.).

Quand ? Les patches sont appliqués automatiquement chaque jour, à une heure précise que vous pouvez définir. Un délai aléatoire entre les différents clients permet d'éviter des connexions simultanées sur le serveur WSUS. Les patches sont aussi appliqués au moment du démarrage, si l'heure spécifiée est dépassée.

Où ? Les patches sont appliqués sur chaque ordinateur ayant le client Automatic Updates configuré. Un *pull* HTTP est utilisé afin de récupérer les mises à jour depuis le serveur WSUS.

Comment ? Les mises à jour sont d'abord approuvées par l'administrateur. Ensuite, les correctifs sont transférés entre le serveur WSUS principal et les serveurs WSUS secondaires, puis vers les clients, en tâche de fond, en optimisant la bande passante grâce au service BITS. Puis les patches sont automatiquement appliqués sur chaque ordinateur, en tâche de fond ou non. Si nécessaire, un unique redémarrage est effectué. Enfin, les journaux de synchronisation et d'approbation (au format XML) sont mis à jour.

WSUS offre des points de distribution multiples pour les patches Windows. La réplication entre les serveurs WSUS internes est également basée sur les services « Automatic Updates » et BITS, permettant d'utiliser la bande passante non utilisée du réseau. Malgré ceci, le transfert des mises à jour peut être très long

Tableau de synthèse des outils de gestion de correctifs proposés par les éditeurs

OS	Outils de scan des correctifs	Outils de gestion des correctifs
SUN Solaris	`pkginfo`	`patchadd`, `smpatch`, `install_cluster`, *Sun Update Manager*, *Solaris Patch Manager*, *JumpStart*
HP-UX	*Security Patch Check Tool* (`security_patch_check`)	*Patch Assessment Tool*, *Custom Patch Manager*
IBM AIX	`lslpp`	`instfix`, `installp`
Linux RedHat		`up2date`, `rpm` (RPM *Package Manager*), RHN (*Red Hat Network*)
Linux Debian		`dpkg`, `apt-get` (*Advanced Package Tool*)
Windows	MBSA 2.0	Microsoft Update, WSUS, SMS
Oracle		Patch Wizard, AutoPatch

(parfois plusieurs jours !). Il faut en effet préciser qu'un jeu complet de patches pour un système tournant sous Windows 2000/XP/2003 représente actuellement environ 1 Go par langue gérée.

Systems Management Server

SMS 2.0 et SMS 2003, qui sont destinés à l'origine à la gestion de parc (inventaire, télédistribution d'applications, etc.) peuvent être utilisés seuls ou bien en combinaison avec des outils complémentaires proposés par Microsoft, spécialisés dans la distribution de correctifs de sécurité.

En effet, WSUS peut s'intégrer à SMS 2003 SP1 grâce à l'outil ITMU (*Inventory Tool for Microsot Updates*), qui est gratuit et permet de déterminer le niveau de patches des systèmes distants. Cet outil supporte les correctifs de Windows Update et Microsoft Update et est capable de les distribuer sur les systèmes qui le nécessitent (voir liens à la fin de cet article).

Par ailleurs, même sans ITMU, vous pouvez utiliser SMS 2.0 et SMS 2003 pour déployer des correctifs de sécurité, même si ce n'est pas l'objectif initial de SMS. Pour vous faciliter la vie, vous pouvez par exemple créer un paquet SMS « lanceur de correctifs intelligent » et ne nécessitant pas de recompilation à chaque ajout de correctif. Ajouter un correctif peut ainsi se réduire à compléter un fichier .INI décrivant ses pré-requis, ses paramètres de ligne de commande et les informations de contrôle de bonne exécution.

De plus, dans un tel paquet SMS, vous pouvez en profiter pour inclure d'autres correctifs applicatifs, comme la mise à jour de l'antivirus des postes de travail, par exemple.

Environnement Unix

Dans l'environnement Unix, chaque éditeur a développé ses propres mécanismes et ses propres outils de gestion des correctifs. De plus, de grands acteurs de la gestion de parc, comme Tivoli, BMC, Novell ou Computer Associates, mais aussi des spécialistes de la gestion de correctifs comme Criston, Landesk, Altiris, RippleTech, Ecora Software, Tenable, Shavlik, PatchLink et St Bernard Software, proposent des solutions intéressantes, car hétérogènes et intégrées.

Ces outils utilisent en général les mêmes principes que ceux présentés précédemment : analyse du niveau de patches, téléchargement des correctifs de sécurité, validation, distribution, vérification d'application correcte et enfin génération de rapports.

Voici un tableau de synthèse des outils (Unix, mais aussi Microsoft et Oracle) proposés par les éditeurs, le plus souvent gratuitement. Certains sont manuels (`apt-get`), d'autres automatisables (RHN par exemple) : tableau ci-dessus.

Conclusion

Comme nous l'avons vu, aucune entreprise ne peut aujourd'hui s'affranchir de la mise en place d'une politique efficace de gestion des correctifs de sécurité. Si de nombreux outils existent, leur mise en œuvre nécessite une véritable réflexion globale afin de limiter le temps d'exposition à une faille tout en ne pénalisant pas l'activité de l'entreprise.

Il faut noter que les éditeurs semblent œuvrer désormais de façon importante afin de rendre leurs logiciels moins vulnérables (configuration par défaut plus restrictive des OS par exemple) ou exposés (firewall embarqué activé par défaut dans Windows XP SP2, protection de la mémoire contre les *buffer overflows* dans Windows 2003 et Windows XP SP2), tout en facilitant encore la gestion des correctifs (avec WSUS notamment).

Cependant, de nouvelles formes de « *hacking* automatisé », toujours plus performantes, apparaissent régulièrement, rendant la veille plus que jamais indispensable à l'anticipation des problématiques de sécurité de demain.

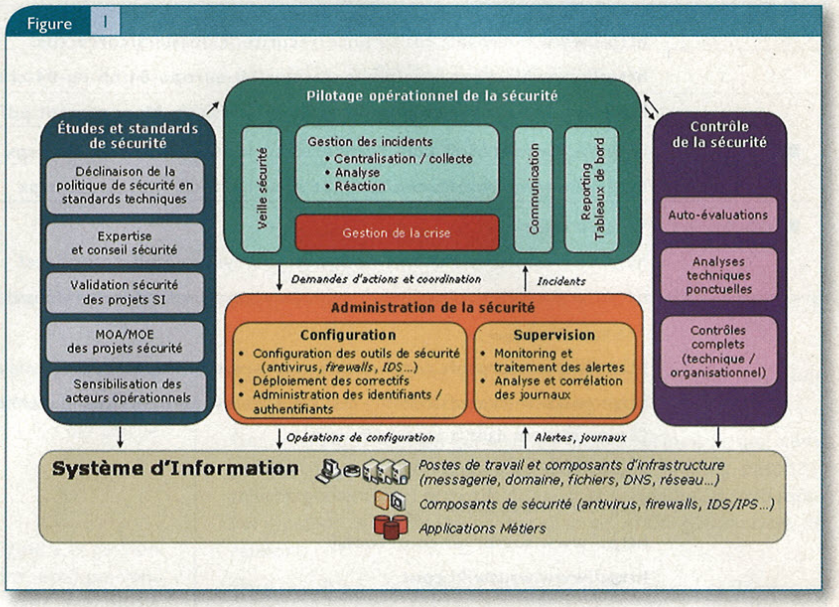
Pour en savoir plus

- Méthodologie pour un processus de gestion des correctifs :
  http://www.giac.org/practical/GSEC/Daniel_Voldal_GSEC.pdf
- Gestion des correctifs Microsoft :
  http://www.microsoft.com/france/securite/it/dossiers/correctifs/
  http://www.blackhat.com/presentations/bh-europe-04/bh-eu-04-chambet-larcher.pdf
  http://www.chambet.com/publications/ISB-Patch_Management.pdf
- MBSA : **http://www.microsoft.com/technet/security/tools/mbsahome.mspx**
  http://www.microsoft.com/technet/security/tools/mbsa2/qa.mspx
- WSUS :
  • Page principale :
  http://www.microsoft.com/windowsserversystem/updateservices/
  http://www.microsoft.com/windowsserversystem/updateservices/downloads/WSUS.mspx
  • WSUS Overview :
  http://www.microsoft.com/windowsserversystem/updateservices/evaluation/overview.mspx
  http://www.microsoft.com/france/securite/evenements/journeesMicrosoftSecurite2005.mspx
  (2 présentations dans la page, en français)
  • Articles et outils utiles :
  http://www.wsus.info
  http://www.susserver.com/Tools/
  http://www.wsuswiki.com
- SMS : **http://www.microsoft.com/smserver/default.mspx**
  ITMU : **http://www.microsoft.com/smserver/downloads/2003/tools/msupdates.mspx**
- MBSA 2.0 : **http://www.microsoft.com/technet/security/tools/mbsahome.mspx**
  http://www.microsoft.com/technet/security/tools/mbsa2/qa.mspx
- SUN :
  • *Solaris Patch Management: Recommended Strategy* :
  http://docs-pdf.sun.com/817-0574-12/817-0574-12.pdf
  http://www.sun.com/blueprints/0205/819-1002.pdf
  • *Update Manager* :
  http://docs.sun.com/source/835-0620/index.html
  • *Patch Manager* :
  http://www.sun.com/service/support/software/patchmanagement/patchmanager.html
  http://sunsolve.sun.com/
  http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-license&nav=pub-patches
- HP : *Patch Management User Guide for HP-UX* :
  http://docs.hp.com/en/5991-1163/index.html
  http://itrc.hp.com
- IBM : **http://www-1.ibm.com/servers/eserver/support/pseries/aixfixes.html**
  http://techsupport.services.ibm.com/server/criticalfixes3/criticalfixes.html
- Linux RedHat : **http://www.redhat.com/software/rhn/**
  http://www.rpm.org/
- Linux Debian : **http://www.apt-get.org/**
- Oracle : **http://www.oracle.com/technology/products/applications/upgrade_patching/**

Reporting et procédures de réaction

L'entreprise est maintenant relativement consciente des risques que court son système d'information. Elle a par conséquent défini de manière plus ou moins formelle une politique de sécurité et mis en œuvre les solutions techniques qui en garantissent l'application. Dès lors la fonction de supervision de la sécurité tend à prendre un rôle central dans les stratégies de sécurité des entreprises. Une des fonctions principales de l'équipe de supervision est le pilotage opérationnel de la sécurité qui définira notamment les procédures de réaction. Chaque action menée par cette équipe viendra alimenter des tableaux de bords opérationnels dont l'analyse permettra de suivre le niveau de sécurité interne et externe dans le temps, par rapport à des objectifs prédéfinis.

Figure 1



Introduction

Le besoin d'automatiser la supervision de la sécurité se justifie généralement par la constante évolution des risques qui pèsent sur le Système d'Information des entreprises. L'évolution la plus remarquée des incidents subis par les entreprises durant les dernières années concerne les menaces liées aux codes malveillants (virus, vers, chevaux de Troie, *spywares*...). Elle s'illustre notamment sur différents axes :

1. **Quantité** : avec une très forte croissance du nombre de codes malveillants en circulation.
2. **Rapidité d'exploitation** des nouvelles vulnérabilités.
3. **Rapidité de propagation** des codes malicieux.
4. Par l'**ingéniosité,** notamment au travers des techniques de « *social engineering* » déployées par les nouveaux codes malveillants.

L'autre évolution marquante est la professionnalisation et la criminalisation de la menace informatique. La professionnalisation se constate à travers la complexité des nouvelles attaques et à travers les méthodes et moyens mis en œuvre. La criminalisation se manifeste par le changement des objectifs des « pirates », qui passent du simple plaisir intellectuel ou du défi technique à des objectifs commerciaux ou stratégiques. À cette professionnalisation de la menace informatique doit donc répondre une professionnalisation de la défense du Système d'Information.

Les missions qui composent cette gestion opérationnelle de la sécurité peuvent être découpées en quatre grandes fonctions :

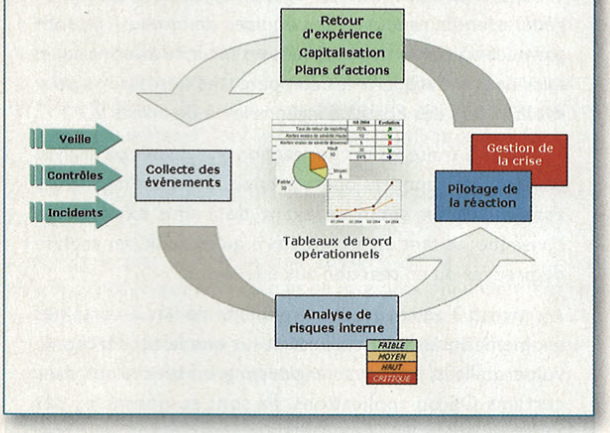
1. Les **études et standards de sécurité**, dont l'objectif est de décliner la politique de sécurité dans les processus projets et les processus d'exploitation.
2. Le **contrôle de la sécurité**, dont l'objectif est de mesurer le niveau de sécurité du Système d'Information et de contrôler la mise en œuvre des actions correctrices.
3. L'**administration de la sécurité**, qui regroupe l'ensemble des actions de configuration et de supervision du Système d'Information.
4. Le **pilotage opérationnel de la sécurité**, qui représente la tour de contrôle en termes de coordination des actions de sécurité au quotidien, tant de manière proactive que de manière réactive.

(voir figure 1, ci-dessus)

Chaque fonction de la supervision de la sécurité permet ainsi d'inscrire la gestion opérationnelle de la sécurité dans une véritable **boucle de progrès**, au cœur de laquelle se situe le **reporting**. L'alimentation et l'analyse de tableaux de bord opérationnels permettent de suivre le niveau de sécurité dans le temps, par rapport à des objectifs prédéfinis. Les tableaux de bord constituent ainsi l'**instrument de mesure de la qualité de la boucle de progrès du dispositif de gestion de la sécurité**.

Sylvain Roger – **sylvain.roger@solucom.fr**
Consultant Sécurité des Systèmes d'Information/SoluCom, **http://www.solucom.fr**
Renaud Bidou – **renaudb@radware.com**
Consultant Sécurité Europe/Radware, **http://www.radware.com**

Figure 2



Les informations collectées par le SOC (*Security Operation Center*) sont une source précieuse d'alimentation de ces tableaux de bord (figure 2).

Chaque opération réalisée par le SOC doit être formalisée par un rapport et doit suivre des procédures spécifiques. Nous traiterons chaque type d'opérations et le couple rapport/procédure dans des parties distinctes. Trois types d'opérations peuvent être distingués :

- les **opérations temps réel**, qui correspondent aux actions qui doivent être menées dans un délai limité ;
- les **opérations tactiques**, effectuées de manière régulière et issues d'une analyse des informations sur une période courte à moyenne ou réalisées en escalade suite au résultat d'opérations en temps réel ;
- les **opérations stratégiques**, réalisées suite aux analyses sur le long terme et destinées aux responsables de la sécurité, aux DSI et à la direction de l'entreprise.

1. Les opérations en « temps réel »

Notion de « temps réel »

D'une manière générale le temps réel est une notion galvaudée par l'ensemble des acteurs du monde de l'informatique et nous n'échappons pas à la règle. Dans le cadre d'un SOC, cette notion est à considérer comme la définition d'une borne supérieure au délai d'exécution d'une action, le délai étant à considérer comme suffisamment court pour apporter une réponse appropriée à une menace visant le système d'information.

Ainsi, avant toute autre chose, il s'agit d'évaluer les différents critères temporels qui correspondent aux actions à mener en réaction à un événement. Cette étape est indispensable, tant d'un point de vue technique que contractuel. En effet, si une erreur intervient/se produit lors de la mise à jour d'une base de signatures d'un IDS, un délai de réaction borné à une heure est acceptable. Dans le cas de la propagation d'un ver, le filtrage des flux doit être effectué dans les 5 minutes à tous les niveaux de l'infrastructure. Passé ce délai, le filtrage devient inutile…

L'aspect contractuel, quant à lui, est à considérer aussi bien pour les SOC intégrés à l'entreprise que pour les SOC externalisés. Dans tous les cas, l'entité s'est vue affecter une mission et doit être à même de rendre des comptes. Cela signifie que des métriques doivent être établies et validées par les deux parties. Le délai maximum est une de ces métriques critiques ! L'ensemble des métriques définies dans le cadre des opérations d'un SOC établissent le SLA (*Service Level Agreement*) dont le non-respect peut entraîner des pénalités financières par exemple dans le cas d'un service externalisé. Le pilotage de ces SLA est possible par les rapports stratégiques (voir plus loin) qui constituent leur principal outil de contrôle (SLM – *Service Level Management*)

Les alertes

Nature des alertes

Les opérations effectuées en « temps réel » sont déclenchées à la suite de la réception d'un événement. Ces événements peuvent avoir deux sources possibles :

Source externe : il s'agit de la veille réalisée en interne par une équipe dédiée (habituellement le niveau 3, soit le plus haut niveau d'expertise) ou par un infogérant (type MSSP). Les sources externes sont responsables, par exemple, de l'identification d'une vulnérabilité dont l'exploitation peut avoir des conséquences importantes pour l'entreprise et pour laquelle un correctif doit être déployé. Il peut également s'agir de l'identification d'une forte activité virale qui nécessite un déploiement en urgence de signature antivirale.

Source interne : Les sources internes sont constituées des éventuels outils de *monitoring* mis en place, tels que les outils de supervision réseau, les consoles centrales d'administration de produits comme les antivirus ou les sondes IDS/IPS correctement configurées pour remonter des informations pertinentes. Une autre source interne, cette fois-ci non technique, est bien évidemment l'utilisateur, qui par l'intermédiaire du *helpdesk* disponible en 24x7 pour les grandes entreprises peut remonter des comportements anormaux.

Qualification des alertes

Une des caractéristiques qui affectent de manière notable la qualité des réactions est la manière dont les alertes sont rapportées au SOC. Si les messages sont précis (pas nécessairement détaillés, précis) et si la criticité de l'événement est évaluée de manière

fiable, alors le phénomène peut être rapidement qualifié et la procédure appropriée lancée dans des délais records.

Les points clefs d'une bonne qualification des messages sont les suivants :

- Le message doit être explicite : « Erreur d'authentification » est préférable à « %SNMP-3-AUTHFAIL » ;
- Les éléments clefs du message (voir l'article sur la collecte) doivent être présentés de manière lisible : « Utilisateur : root » vaut mieux que « uid = 0 » ;
- L'agrégation de messages identiques est obligatoire : prenons le cas du lancement automatique d'exploits divers et variés, détectés par un IDS. Un unique message du type « Lancement d'exploits » doit être envoyé au SOC, complété par une information en temps réel précisant le nombre d'exploits est disponible ;
- La criticité doit être visible et son calcul fiable : l'évaluation de la criticité d'un événement est une des opérations les plus difficiles dans la mesure où elle prend en compte, d'une part, l'importance de la ressource cible et, d'autre part, l'impact de l'attaque sur cette ressource. Un autre critère intéressant est le stade de l'intrusion auquel correspond l'événement, qui ne peut être défini qu'à partir de scénarii (voir plus loin).

Il est également nécessaire de faire attention à la présence de multiples alertes (réduites normalement par l'agrégation) et d'éventuels faux positifs. En effet, dans ce type de schéma où plusieurs sources sont utilisées pour superviser les événements de sécurité qui se produisent sur le système d'information de l'entreprise, plusieurs alertes peuvent être générées pour un même incident de sécurité et aboutir à l'ouverture de multiples tickets d'incidents par le support de niveau 1. Les outils de corrélation de type SIM ont pour rôle (entres autres) de réduire la probabilité que de tels cas se produisent. Enfin, les équipements installés dans le cadre de la supervision interne peuvent générer des alertes en réponse à des événements conformes (se reporter au précédent dossier sur les limites de la sécurité). Ces fameux faux-positifs ont pour conséquence de surcharger les différents niveaux de support (support niveau 1 par l'ouverture de ticket d'incident intempestif, supports de niveau 2 et 3 par la consommation de temps et d'expertise pour l'analyse d'un incident qui n'en est pas un) et surtout de provoquer des réactions non justifiées, ce qui est relativement grave quand cela aboutit à couper un flux ou arrêter un serveur...

Réaction

La génération des alertes ne doit être une fin en soi. Ces dernières doivent être produites à bon escient afin de fournir la meilleure réponse. C'est pourquoi il est nécessaire de s'appuyer sur l'architecture organisationnelle mise en place au sein du SOC (cf. premier article du dossier de ce numéro) pour optimiser la réaction. Cette dernière doit reposer sur des procédures prédéfinies et rôdées, connues par l'ensemble des acteurs.

Acteurs

Les destinataires des alertes sont les membres d'une équipe qualifiée de niveau 1. C'est à eux que revient la lourde mission de lancer les procédures de réaction adaptées à chaque événement.

Les niveaux du SOC

Trois niveaux d'intervenants sont définis dans un SOC, correspondant chacun à un niveau d'escalade. Le niveau 1 est composé de techniciens dont le rôle principal est d'appliquer à la lettre les procédures établies, testées et validées en amont. Il peut s'agir d'opérations de réaction ou d'opérations régulières de contrôle ou de génération de rapports. Les équipes de niveau 1 ne sont pas nécessairement spécialisées en sécurité informatique, mais peuvent acquérir les compétences nécessaires pour évoluer vers des postes d'ingénierie ou de niveau 2.

Le niveau 2 traite les cas de réaction non définis ou analyse en détail les rapports opérationnels. Il s'agit d'ingénieurs spécialisés en sécurité ayant déjà une expérience conséquente tant en architecture qu'en audit, en analyse de preuves ou en réaction aux intrusions.

Au niveau 3 se trouvent les experts dédiés à certaines problématiques particulières telles que la recherche de vulnérabilités, le *reverse engineering* ou spécialisés dans certains OS ou applications. Ils sont responsables des opérations de R&D et d'analyse poussée de menaces particulières, telles qu'un 0-day ou un nouveau ver.

Cependant, il ne s'agit en aucun cas d'experts en sécurité. S'il est évident qu'un léger *background* dans le domaine est un plus certain, cela reste une caractéristique « *nice to have* ». Le critère principal est la rigueur et la capacité de résistance au stress.

La rigueur est indispensable dans la mesure où le respect des aspects contractuels n'est envisageable que si les procédures définies sont appliquées rapidement et à la lettre. Le niveau 1 du SOC n'a pas vocation à effectuer des fonctions de recherche et/ou d'analyse. Dès lors, la pertinence de la qualification des alertes prend tout son sens. En ce qui concerne la résistance au stress, il semble que l'explication est évidente...

Les actions du SOC

D'une manière générale le niveau 1 du SOC est amené à effectuer deux types d'opérations :

- une action immédiate visant à circonscrire la menace ;
- une escalade dans le cas d'un événement pour lequel aucune procédure n'est définie ou pour les actions ne nécessitant pas une réaction en temps réel.

Dans le premier cas, le schéma est trivial :

1. L'alerte est reçue ;
2. La procédure de réaction est identifiée en fonction d'un ou plusieurs critères parmi les suivants : « type d'événement », « cible », « ressource » ou « criticité » ;
3. Les actions prédéfinies sont appliquées ;
4. Le ticket est clos.

L'élément clef est ici le point 3. Il implique que les procédures de réaction ont été définies, testées et validées en amont, ce de la manière la plus exhaustive possible. C'est à cette unique condition que le SOC pourra garantir une réaction appropriée dans des délais prédéfinis.

L'escalade est utilisée dans deux cas. Le premier est la réception d'une alerte pour laquelle aucune réaction n'a été définie. Dans ces conditions l'équipe de niveau 1 ne peut prendre la responsabilité de réagir en temps réel et doit escalader vers une équipe plus spécialisée et non soumise aux mêmes contraintes. Elle doit néanmoins rester en attente de la procédure de réaction qui lui sera fournie par le niveau 2 et qui, une fois appliquée, permettra de clore le ticket. Le second cas d'escalade est tout simplement lié à des événements dont le traitement s'effectue dans la durée ou nécessite la mise en place de procédures complexes telles que la gestion des correctifs sur les 2.000 postes de travail de l'entreprise.

Dans un dernier temps, les opérations « temps réel » réalisées devront faire l'objet d'un suivi et de rapports afin d'identifier des tendances à moyen terme des menaces touchant l'entreprise et du niveau de protection actuellement mis en place. Ce reporting permettra de mener des opérations plus tactiques.

2. Les opérations tactiques

Principe des opérations tactiques

Au niveau tactique, il s'agit de fournir des informations pertinentes en termes d'analyse a posteriori. Ce sont des informations techniques dont l'objectif est d'aider à la compréhension de phénomènes complexes, faciliter les opérations d'investigation et fournir les éléments nécessaires aux choix d'évolution de l'infrastructure de sécurité.

Par conséquent, il faut considérer deux caractéristiques fondamentalement différentes des opérations tactiques. La première est une analyse programmée d'évaluation des risques sur un périmètre (infrastructure, menace, temps) défini, la seconde est l'investigation des causes d'une opération en temps réel. Dans le premier cas, nous trouverons les passionnantes opérations consistant à comparer les résultats des *scans nessus* de cette semaine et de la semaine précédente, l'inventaire des virus les plus filtrés par la passerelle ou encore l'exaltante charge de vérifier l'état des mises à jour de Windows sur les postes de travail des secrétaires.

Les opérations d'investigation ont quant à elles pour but de remonter les informations liées à un événement afin de comprendre le contexte et de mettre en œuvre les opérations de correction ou de réponses nécessaires. Identifier la source d'un ver qui se propage à vitesse grand V sur le réseau, étudier la cause (et les conséquences) d'anomalies de trafic sur le réseau de VoIP ou encore analyser la nature de flux sortants bloqués massivement par le firewall la nuit dernière sont des opérations types de cette catégorie.

Les audits post-mortem et l'étude de *forensics* font également partie de ce type d'opérations.Dans ce cas, les opérations tactiques sont typiquement les actions menées par l'escalade de niveau 2 des opérations en temps réel et, en cas de difficulté particulière, une escalade vers le niveau 3 peut s'avérer nécessaire.

Organisation et présentation des données

Quelle que soit l'opération menée, un élément clef est à mettre en avant : la manière dont les données sont présentées. L'identification d'activités latentes, telles qu'un stupide scan de ports effectué sur une durée relativement longue, ou complexes, telles que des attaques impliquant plusieurs sources associées à des tentatives de saturation de logs, implique le traitement de quantités considérables de données.

Faute d'une organisation cohérente de ces données, il est illusoire d'imaginer que le rendu soit convaincant.

Cette organisation doit rendre possible au moins l'une des deux opérations suivantes :

- Tri et regroupement selon un ou plusieurs critères (source, date, type d'événement etc.) ;
- Analyse et détection de scenarii d'attaque.

Tri et regroupement

Dans le premier cas, il s'agit d'être à même d'effectuer des analyses à partir d'un événement caractéristique, ce qui se résume en général à des opérations a posteriori. Suite à la remontée d'un événement caractéristique (par exemple l'arrêt du service syslog sur un serveur), il est nécessaire d'obtenir rapidement une information fiable et concise sur l'ensemble des événements concernant la même cible.

Dans le cas d'une attaque simple, c'est-à-dire effectuée dans une fenêtre de temps suffisamment courte et à partir d'une source unique, un simple filtre sur la source, la cible et la fenêtre de temps en question permettra d'identifier rapidement la succession d'événements qui ont mené à la compromission.

Un tel schéma est représenté par exemple par le regroupement suivant :

Destination : 10.0.0.1 – Source : 192.168.0.1
Heure : 10 :00 :00 – 10 :02 :13 : Paquets filtrés (243)
Heure : 10 :02 :17 : Identification d'OS (1)
Heure : 10. :04 :48 : Identification d'application (1)
Heure : 10 :13 :55 : Lancement d'exploit (1)
Heure : 10 :15 :27 : Arrêt de service (1)

Imaginons maintenant qu'il s'agisse d'une opération un petit peu plus subtile : un stagiaire malveillant et patient, chargé de récupérer quotidiennement les bandes de sauvegarde dans la salle informatique, essaie de se connecter à la console du serveur à chaque fois qu'il est dans la salle informatique. Les éléments caractéristiques permettant d'identifier cette action sont a priori : les erreurs d'authentification, une ouverture de session et l'arrêt du service syslog.

L'analyse lancée sur une période restreinte « autour » de l'événement déclencheur (arrêt du service) ne remontera que :

Destination : 10.0.0.1 – Source : Local
Heure : 10 :00 :00 – 10 :00 :00 : Ouverture de session (1)
Heure : 10 :02 :17 : Arrêt de service (1)

Autant dire que l'analyse risque d'être rapide… Si maintenant le critère temps est supprimé et que le regroupement est réalisé par Cible/Source/Type d'événements, le schéma deviendra tout de suite plus clair :

Destination : 10.0.0.1 – Source : Local
Erreur d'authentification (129)
Ouverture de session (1)
Arrêt de service (1)

La deuxième étape de la traque, telle qu'elle se déroulerait dans le meilleur des mondes, suppose que l'accès à la salle informatique est restreint, que la « badgeuse » envoie un message à un serveur, et que celui-ci est traité par le SOC. S'il est possible de générer ce rapport avec comme « cible » de l'attaque le système en 10.0.0.1 et l'infrastructure physique, en limitant les messages à « accès à la salle informatique » et « erreur d'authentification », le schéma final (intégrant les aspects temporels cette fois) nous donnera quelque chose comme ça :

Destination : 10.0.0.1 ou Salle informatique – Source : Local
Jour 1 / Heure : 10 :00 :00 : Accès salle (1)
Jour 1 / Heure : 10 :02 :17 : Erreur d'authentification (1)
Jour 2 / Heure : 10 :07 :18 : Accès salle (1)
Jour 2 / Heure : 10 :13 :14 : Erreur d'authentification (1)

Etc. et Gotcha ! Bien entendu cela implique que la position du serveur soit connue (dans la salle informatique), ce qui est généralement intégré aux outils de gestion de réseau via des modules d'inventaire, également utilisés pour le support et la maintenance.

L'approche par tri et regroupement est également à privilégier dans le cadre des opérations tactiques de fond. L'apparition de nouvelles vulnérabilités, une activité virale particulière suite à l'apparition de la 264ème variante de Zotob ou encore le statut des mises à jours se prêtent particulièrement bien à une représentation organisée selon les axes « date », « type d'événement » (ici « *update successful/failed* », « virus filtré » ou encore « nouvelle vulnérabilité détectée » et « ressource » (permet d'identifier le virus, la version de logiciel, etc.).

Jour 1
Mise à jour réussie : Anti-virus pattern v.12345 (432)
Mise à jour échouée : Anti-virus pattern v.12345 (7)

La seconde étape sera de demander le détail des « cibles » sur lesquelles la mise à jour a échoué, rapport qui est obtenu via un simple filtrage sur les champs « type d'événement » et « ressource ». Cet exemple est intéressant dans la mesure où il pose la question de l'application de la politique de sécurité aux postes nomades en dehors du réseau de l'entreprise. En effet, une telle information (succès ou échec de la mise à jour) ne peut être obtenue que si l'anti-virus trouve son serveur et est à même de transmettre l'information au serveur d'analyse d'événements, ce qui nous mène directement à l'obligation de connexion au réseau de l'entreprise via un VPN.

Analyse par scénarii

L'approche par scénarii consiste à établir une séquence logique d'événements caractéristique d'une tentative d'intrusion. Elle présente un inconvénient majeur : les « *hackers* » se refusent à suivre les étapes que l'on a définies pour eux ! Par conséquent, il ne faut pas attendre que les moteurs de corrélation géniaux proposant ce type d'analyse fassent le travail des ingénieurs en charge des opérations tactiques. Cependant, pourvu qu'ils soient bien configurés (ce qui est le deuxième gros écueil après l'obtention du budget pour leur acquisition), ils sont à même d'avoir une fonction préventive particulièrement intéressante. En effet, si le fait de générer une alerte lorsqu'un scénario est complété ne mènera en général à rien d'autre qu'à la détection de scans de ports, il s'avère beaucoup plus intéressant de générer une alerte lorsque *n*% du scénario est complété/achevé.

De même, en cas d'utilisation d'un 0-day, il est probable que 90% des étapes du scénario auront été observées, les 10% restant étant liés à l'exploit lui-même. Dans ce cas encore, l'approche par scénarii peut accroître la qualité de la réponse du SOC.

Remontée telle quelle au niveau 1, ce genre d'alerte doit immédiatement être escaladée au niveau 2 pour une analyse tactique immédiate, dans la mesure où elle est probablement caractéristique d'une intrusion en cours. Elle nécessitera cependant une analyse approfondie pour s'assurer qu'il ne s'agit pas d'une fausse alerte et que le SOC ne va pas brutalement interrompre la visioconférence du Directeur France avec le QG aux États-Unis. L'efficacité de la prévention tient donc à deux éléments essentiels :

- la fiabilité des scénarii, qui ne peuvent être construits de manière générique mais doivent intégrer les spécificités techniques et comportementales du SI et de ses utilisateurs ;
- la qualité de la représentation des informations et la rapidité d'accès aux éléments caractéristiques du scénario.

Dans ces conditions, l'analyse par scénarii, couplée à un SOC compétent, organisé et disponible, devient un outil redoutable de lutte contre les activités malveillantes.

Un autre intérêt des scénarii est l'intégration d'informations structurelles, permettant par exemple de coupler une information d'état telle que la vulnérabilité d'un serveur à une attaque et à une information d'action telle la détection de cette attaque à destination du serveur en question. Le principe du scénario est trivial, mais particulièrement efficace si les bases de vulnérabilités et de signatures sont à jour (et compatibles bien entendu) et s'il ne s'agit pas d'un 0-day…

Escalade, réaction et suivi

A l'instar de toutes les opérations menées par le SOC, les opérations tactiques doivent êtres liées à une ou plusieurs actions.

Le premier cas est l'escalade. Cette action intervient lorsque le niveau 2 ne dispose pas des compétences nécessaires pour évaluer la situation. C'est un cas fréquemment rencontré lorsqu'il s'agit d'attaques particulièrement complexes, de 0-day, lorsque certains éléments de détection ont été « évadés » ou lorsque les équipes recrutées pour effectuer ce travail ne sont pas suffisamment compétentes... Faute de pouvoir mener une action préventive ou correctrice, le niveau 2 doit escalader au niveau supérieur, censé fournir une expertise technique poussée. La capacité de transfert des informations vers le niveau supérieur est un élément critique dont la qualité est un élément clef pour l'efficacité globale du processus. Si le niveau 3 doit de nouveau effectuer l'ensemble des opérations tactiques, la perte de temps peut s'avérer critique dans le cas de la propagation d'un ver ou d'une action préventive liée à un scénario. Il est donc primordial que l'ensemble des opérations tactiques soit documenté (de manière semi-automatique si possible) et transmis à l'escalade.

En ce qui concerne la réaction, ce n'est théoriquement pas le rôle des équipes tactiques. Si le fruit de leurs investigations conduit à un processus déjà établi, leur analyse doit être transmise au niveau 1. Ce dernier mettra en œuvre l'action prédéfinie. En revanche, s'il s'agit d'un cas non prévu dans les procédures existantes, il doit être exposé (plus ou moins rapidement en fonction de sa criticité) et une procédure de réaction doit être définie, testée, intégrée à l'existant et, bien sûr, appliquée. Concrètement, il est rare, dans l'urgence, que toutes ces étapes soient suivies dans cet ordre. Généralement une solution d'urgence est appliquée (et souvent dans la panique), puis les étapes « normales » sont suivies...

Enfin, il est indispensable que les actions menées soient suivies afin de connaître précisément le statut des actions en cours. Intégrer les outils utilisés pour les opérations tactiques à un système de *workflow* est donc un point particulièrement important. A défaut d'un suivi propre et organisé, il existe des risques importants que les escalades ne fournissent aucun résultat ou qu'une fois une solution d'urgence mise en place la procédure propre et officielle ne soit jamais écrite.

Gestion de tickets d'incidents

Le traitement organisé d'événements de sécurité, que ce soit à l'échelle tactique (« temps réel ») ou opérationnelle s'effectue selon les mécanismes classiques de gestion des tickets d'incident. Par conséquent une personne doit être responsable de chaque action (prendre le *ownership*), des escalades automatiques doivent être programmées en cas de dépassement de certains délais, les actions menées doivent être enregistrées et la fermeture d'un ticket d'incident doit intégrer la mise à jour des interfaces.

Les outils génériques de gestion des tickets d'incidents sont tout à fait adaptés à la gestion des opérations de sécurité, dans la mesure où les outils de collecte et d'analyse sont à même de s'intégrer.

3. Les opérations stratégiques

Les rapports stratégiques

Qu'est-ce qu'un rapport stratégique ?

Les rapports stratégiques ont pour objectif de fournir une vue macroscopique de l'activité sécurité d'un système d'information. Il s'agit par conséquent d'informations fondées sur le long terme, ce qui a un nombre certain d'implications techniques, car il va falloir être à même de traiter et synthétiser un nombre important de données.

La deuxième caractéristique des opérations stratégiques est leur portée. Cette dernière est toujours large, que ce soit d'un point de vue fonctionnel, géographique ou organisationnel. Ainsi, les informations fournies porteront par exemple sur l'activité virale sur les postes de travail des différentes filiales, les principales sources géographiques des attaques externes ou encore les taux de propagation des vers ayant impacté le système d'information de l'entreprise.

Enfin, les données contenues dans ces rapports doivent être très synthétiques, généralement concentrées sur un ou deux graphes et ne doivent en aucun cas être techniques.

A quoi servent les rapports stratégiques ?

Les rapports stratégiques sont théoriquement la base nécessaire à des prises de décision de haut niveau, qu'il s'agisse de définir ou d'affiner la politique de sécurité, de planifier de nouveaux investissements ou encore d'établir une stratégie d'intégration d'un autre SI, à l'issue d'une fusion par exemple. Leur fonction première est donc d'être un outil d'aide à la décision et ils sont un élément clef du pilotage global de la sécurité du système d'information.

Concrètement, ces rapports servent également à justifier la politique de sécurité et les budgets qui lui ont été alloués. En effet, les infrastructures coûtent chers et les politiques de sécurité sont impopulaires. Il est généralement difficile de les faire évoluer sans fournir d'éléments justificatifs concrets et compréhensibles par des directeurs généraux et des directeurs financiers. Ainsi, lorsqu'un rapport stratégique montre des pics de tentatives d'accès échoués sur l'intranet, il devient beaucoup plus simple de faire passer un projet d'authentification forte, qui va coûter cher, être long à déployer et va embêter tout le monde...

Dans le même ordre d'idée, ces rapports vont fournir aux responsables de la sécurité un moyen de justifier les investissements effectués dans les infrastructures. En effet, après avoir obtenu (et dépensé !) un budget systématiquement considéré comme important par les directions, les RSSI doivent pouvoir les justifier à partir de critères mesurables. Il est impensable qu'une phrase laconique précisant qu'il ne s'est rien passé et que donc l'investissement était justifié suffise. En revanche, un rapport stratégique montrant qu'en moyenne 1.500 attaques sont bloquées chaque jour et détaillant la répartition des cibles en fonction des filiales est un accessoire indispensable.

Force est de reconnaître que les deux dernières utilisations des rapports stratégiques sont destinées à de basses utilisations pécuniaires. Néanmoins, il faut garder à l'esprit que l'argent est le nerf de la guerre et que la gestion des contraintes politiques et financières est part entière du travail d'un responsable de la sécurité, qu'on le veuille ou non.

3.000.000 de logs par jours et moi et moi

Il apparaît comme évident que la principale problématique rencontrée pour la mise en place d'un plan d'action cohérent et efficace réside dans la manière dont les données vont être représentées. Bien entendu, la représentation dépend tout d'abord de la finalité du rapport et par conséquent du rôle du SOC. Aussi, est-il vain de tenter de donner le contenu type d'un tel rapport, au même titre que de fixer une périodicité des revues, voire de définir les personnes auxquelles il doit être présenté.

Un autre point important à prendre en compte est l'ensemble des réglementations liées à la conservation des données. En effet, autant la loi sur la sécurité quotidienne impose à certains acteurs de conserver les données, autant la CNIL limite la quantité d'informations contenue et autorise un droit de rectification. Il est donc nécessaire de considérer les aspects de volume, de chiffrement, de recherche de données personnelles, le tout avec la gestion des droits appropriée…Outil d'aide à la décision.

Prendre une décision stratégique concernant la sécurité du système d'information n'est pas une action anodine. En effet, accroître le niveau de sécurité de zones particulières ou lancer le déploiement de nouvelles technologies pour faire face à des menaces émergentes sont des opérations qui vont nécessairement impacter le fonctionnement du système d'information, mobiliser des ressources considérables et, encore une fois, coûter très cher. Il convient par conséquent de s'appuyer sur des données factuelles et fiables.

Dès lors, comment organiser les données pour faire ressortir les éléments clefs ? Une erreur commune est de faire l'acquisition d'un outil et de « voir » ce qu'il sait faire.

En effet, cette démarche présente deux défauts majeurs :

1. Chaque entreprise connaît des problématiques différentes. Qu'il s'agisse de la sensibilité à la sécurité, de son organisation interne, de la politique mise en place ou encore de son degré de maturité technologique.

2. Elle correspond à la démarche inverse, souvent rencontrée, il faut le reconnaître, qui consiste à choisir un outil avant d'avoir précisément défini les besoins.

D'une manière plus rationnelle, il faut appliquer une démarche quasi similaire à celle théoriquement utilisée pour la définition d'une politique de sécurité. D'abord identifier les actifs puis les critères du DICAP (Disponibilité, Intégrité, Confidentialité, Auditabilité et Preuve) qui leur sont associés. La matrice obtenue représente l'armature des informations qui doivent apparaître dans le rapport.

Sur cette ébauche, il faut alors plaquer les critères organisationnels et techniques de l'entreprise. De tels critères sont par exemple la répartition géographique des sites sur lesquels sont hébergés ces actifs, les différentes directions (nationales, régionales, systèmes, réseau, etc.) dont ils dépendent ou encore la zone de sécurité à laquelle ils appartiennent. L'ensemble de ces critères permet de définir les filtres à appliquer sur les données.

Une fois ces filtres définis, il ne reste plus qu'à déterminer les axes des différents graphes qui en découleront. Le temps est un axe très fréquemment retrouvé, car il permet d'évaluer une tendance. Le nombre d'événements (et les différents calculs tels que la moyenne mobile) est également une valeur très populaire au même titre que la criticité des événements, les sources et les cibles des attaques. Concernant ces deux derniers axes, il s'agit généralement de groupes dans la mesure où un actif est rarement représenté par un seul système, mais est lié à une chaîne applicative. L'identification d'une cible comme 10.0.0.1 est souvent inutile à ce niveau de décision. En revanche, savoir que c'est la plate-forme de VoIP qui est le plus victime d'attaques venant de l'intranet est une information beaucoup plus pertinente.

Une fois ces éléments définis, il ne reste plus qu'à choisir l'outil, ce qui devrait être alors relativement simple, puisque l'on sait ce que l'on veut…

Actions financières

Comme nous l'avons vu précédemment, les aspects financiers sont parties intégrantes de la stratégie de sécurisation du système d'information et, à ce titre, le SOC peut et doit fournir des informations adaptées. Typiquement, ces données sont tirées des rapports d'aide à la décision, avec un niveau d'abstraction supérieur, une couverture plus générale, des couleurs et de la 3D.

L'exercice est relativement simple lorsqu'il s'agit de justifier un investissement dans la mesure où il suffit de présenter de manière récurrente les mêmes données mettant en avant les avantages des technologies ou ressources déployées. Il s'agira par exemple de fournir chaque trimestre l'évolution du temps de réponse du SOC à un incident pour justifier de l'embauche de nouveaux ingénieurs ou la répartition des attaques sur l'année écoulée, justifiant la mise en place d'un IPS sur la DMZ, etc.

En revanche, l'obtention d'un budget nécessite généralement de prendre en compte des données qui n'étaient pas intégrées ou du moins pas mises en avant dans les procédures de reporting. Par conséquent, il va falloir non seulement redéfinir un rapport (ici il ne s'agit généralement que d'un graphe) qui démontre clairement les différentes problématiques rencontrées au cours du temps et leur impact potentiel.

Ce dernier point est le plus délicat. En effet, prouver que des pics d'erreur d'authentification sur le réseau interne en pleine nuit sont caractéristiques d'une tentative d'accès illégitime est une chose. Expliquer que le succès d'une telle opération peut coûter tant de millions d'euros en est une autre. Pire, il se peut que faute d'attaque fructueuse, la réponse soit simplement que la sécurité en place a tenu et que donc elle est suffisante. Il convient par conséquent de présenter les faits, la probabilité pour qu'une telle attaque soit fructueuse et enfin le coût pour l'entreprise du succès d'une telle attaque. C'est de la gestion de risque, l'opération la plus stratégique qu'un responsable de la sécurité soit amené à effectuer.

Conclusion

Du blocage d'une source scannant les ports d'une manière un peu trop insistante à l'évaluation du coût de la non-sécurité, un SOC est à même de remplir un éventail considérable de missions :

- la réaction en temps réel aux incidents ;
- le maintien du niveau de sécurité de l'infrastructure ;
- l'investigation d'incidents ;
- l'identification des faiblesses de l'architecture ainsi que des éléments les plus exposés ;
- l'analyse de risque.

Cependant, il est encore rare, en entreprise, de voir un département centraliser l'ensemble de ces fonctions, que ce soit pour des raisons de définition du rôle du SOC à sa création, des raisons organisationnelles ou, plus bêtement, des raisons techniques dans la mesure où les outils acquis à grands frais pour la mise en place du SOC ne permettent pas telle ou telle opération, et qu'il faut attendre le budget 2012…

Il n'en reste pas moins que le SOC se doit d'être la clef de voûte de la sécurité du système d'information dans la mesure où il a un rôle fédérateur des événements de sécurité. Qui plus est, il ne faut pas non plus oublier son rôle actif dans la sécurité. Bien organisé, un SOC garantit une meilleure réactivité aux incidents et change peu à peu l'aspect passif de la sécurité.

À l'origine, outil de pilotage et d'aide à la décision, un SOC techniquement bien conçu et proprement organisé peut devenir une arme de prévention d'une remarquable efficacité couvrant l'intégralité du système d'information.

DHIS comme Distributed Hidden Storage

Introduction

Cacher des données de la manière la plus invisible qui soit sans qu'elles puissent être décelées par une quelconque analyse post-intrusion, c'est l'objectif de chaque pirate qui souhaite laisser des données sur une machine qu'il vient de compromettre. La méthode la plus efficace est certes de ne rien écrire sur le disque et d'utiliser la mémoire. Mais cette technique peut vite devenir contraignante : nécessité de mapper lors de chaque accès sur la machine, un réseau très lent, etc. Il n'y a donc parfois pas d'autre solution que de laisser les données physiquement sur la machine.

L'article présente une technique de dissimulation de données avec à l'appui un outil ou plutôt un proof of concept puisque au moment où nous écrivons ces quelques lignes, toutes les fonctionnalités ne sont pas encore implémentées.

Le concept présenté dans cet article utilise le format standard ELF des exécutables et librairies. Nos tests ont donc été effectués sur une machine Linux avec la version 0.65rc1 de l'outil [elfsh] et de sa librairie libelfsh.

Historique

Pour cacher entièrement un binaire (ou un fichier quelconque) sur un système d'exploitation, plusieurs techniques existent déjà. Sous Linux, la technique la plus simple est d'utiliser le noyau. En 1998, [plaguez] a écrit un des premiers articles sur le sujet. L'idée est simplement de modifier un appel système responsable du listing des fichiers. Lorsqu'on liste les fichiers à l'aide de la commande `ls`, celle-ci fait appel entre autres à l'appel système `getdents()`. On peut observer ceci à l'aide de la commande `strace` sous Linux :

```
$ strace ls
execve("/usr/bin/ls", ["ls"], [/* 40 vars */]) = 0
brk(0)                                  = 0x8059790
...
fcntl64(3, F_SETFD, FD_CLOEXEC)         = 0
getdents64(3, /* 102 entries */, 4096)  = 2872
brk(0)
...
$
```

L'appel système `getdents()` va renvoyer une structure contenant tous les fichiers présents dans un répertoire. Il suffit de modifier cette structure pour qu'elle ne contienne pas les fichiers que l'on veut cacher. Pour ce faire, on détourne l'appel système, ce détournement pouvant se faire aussi bien au niveau de la *syscall table*, de l'IDT (*Interrupt Descriptor Table*) ou encore du VFS (*Virtual File System*).

L'inconvénient de ces techniques est qu'il est nécessaire d'avoir accès au noyau pour pouvoir les utiliser. Si ce n'est pas le cas, il existe d'autres techniques pour cacher un fichier qui n'ont pas besoin du noyau. Celles-ci opèrent uniquement au niveau de l'espace utilisateur (*userland*).

Une de ces techniques, [runeFS] développée par The Grugq, utilise un inode particulier, le *bad blocs inode* que nous retrouvons dans la structure du système de fichiers ext2. Celui-ci est utilisé pour référencer les secteurs défectueux d'un disque dur. L'idée est de stocker des données dans les blocs alloués au bad blocs inode. Les secteurs étant marqués défectueux, ils ne seront jamais utilisés pour d'autres données. Cependant, si une analyse de ces secteurs est réalisée, les données cachées peuvent être perdues.

Un autre logiciel, appelé [Hydan], permet aussi de cacher des données dans un binaire. Il remplace certaines instructions par d'autres instructions équivalentes. Par exemple, ces deux instructions sont équivalentes :

```
add %eax, $50
sub %eax, $-50
```

Une convention est ensuite utilisée pour encoder un 0 ou un 1. Par exemple, un `add` représente un 1 et un `sub` représente un 0. Ce logiciel ne permet pas cependant de cacher un binaire entier et est plutôt utilisé pour dissimuler des messages de petites tailles.

Pour ce qui est de cacher des données dans des fichiers ELF, certaines techniques ont déjà vu le jour (NB : hydan peut également cacher des messages dans des fichiers ELF). Il y a quelques années Silvio Cesare a été le précurseur dans ce domaine en créant plusieurs virus ELF ([vit], [siilov]...). Une des premières techniques [elfparasite] consistait à cacher les données dans le *padding* à la fin du segment `text` (situé juste avant le segment `data`). Malheureusement, ce padding ne fait que 4 ko maximum, insuffisant pour y stocker un gros binaire. Même en distribuant l'information, il faudrait pour un binaire de 2 Mo, 500 binaires au minimum pour les cacher. Malgré tout, cette technique reste intéressante pour cacher de tout petits binaires, un *shellcode* ou encore un message.

Dans le même style, une autre technique implémentée dans le virus [siilov] fut d'ajouter les données à la fin du segment `data`.

Le logiciel [ELFSH] permet également de placer des données dans un fichier au format ELF (binaires ou librairies) à plusieurs endroits. Bien que les techniques utilisées dans elfsh soient plutôt orientées détournement du *control flow*, certaines peuvent également s'appliquer pour cacher des données. Pour connaître les dernières techniques utilisées par ELFSH, le lecteur est invité à lire le tout récent article paru dans le dernier *Phrack*.

Toutes ces techniques ont leurs avantages et leurs inconvénients. C'est sur ces inconvénients que nous avons construit notre analyse.

L'idée de départ

L'inconvénient principal de toutes ces techniques est la disponibilité de l'information. Si un expert détecte des données cachées, il récupère la totalité de ces données et, par conséquent,

François Gaspard
kad@miscmag.com
Samuel Dralet
zg@kernsh.org

il peut en comprendre la signification. Au pire, si les données sont chiffrées, il pourra essayer de reverser l'algorithme de chiffrement.

D'une manière générale, l'idée ici est de rendre la détection des données cachées plus difficile en fractionnant l'information et en cachant chaque partie à des endroits différents du système. On distribue en quelque sorte les données cachées. De cette manière, si l'expert suppose avoir détecté quelque chose :

- Il peut au mieux comprendre partiellement l'information qu'il a récupérée, puisqu'il n'a pas tous les morceaux du puzzle.
- Il ne peut pas certifier à 100% que c'est de l'information cachée puisqu'il ne la comprend pas complètement.

Distribuer l'information cachée est l'idée maîtresse dans notre outil. Il faut ensuite savoir comment découper les données, à quel endroit les stocker sur le système et quel type de données cacher.

Rappels sur le format ELF

ELF ou *Executeable and Linking Format* est le format de binaire standard sous Linux. Nous le retrouvons sur d'autres systèmes d'exploitation tels que FreeBSD, NetBSD, Solaris ou encore Irix.

Il est composé d'un en-tête et de sections qui sont référencés dans le fichier `elf.h`. L'en-tête ELF recense les informations sur le format proprement dit (type du binaire, architecture...), la *Program Header Table* (PHT), les segments du binaire, et la *Section Header Table* (SHT) chaque section du binaire.

Nous allons nous focaliser sur la SHT. Chaque section du binaire a en fin de compte son en-tête dans la SHT, chaque entrée étant codée sur sizeof(Elf32_Shdr). Les noms des sections sont stockés dans une section spéciale appelée *Section String Table* (.shstrtab).

D'autres sections composent un fichier ELF :

- les sections de code (.text, .plt, .init, .fini) ;
- les sections de données (.data, .rodata, .bss) ;
- la table des symboles importés et exportés (.dynsym) ;
- la table des noms (.dynstr) et de *hash* (.hash) des symboles dynamiques ;
- la table de relocation (.rel.*) ;
- les tableaux des constructeurs (.ctors) et destructeurs (.dtors) ;
- les sections réservées aux binaires dynamiques (.dynamic, .got, .plt).

Pour référencer chacune de ces sections, on utilise des *offsets* (*foffset* ou *file offset*) depuis le début du fichier ELF.

```
$ elfsh -f /bin/cat -s
created IMMED with val = 0

 [*] Object /bin/cat has been loaded (O_RDONLY)

 [SECTION HEADER TABLE .::. SHT is not stripped]
 [Object /bin/cat]

 [000] 0x00000000 -------                 foffset:00000000 size:00000000 link:00
info:0000 entsize:0000 align:0000 => NULL section
 [001] 0x08048114 a------ .interp         foffset:00000276 size:00000019 link:00
info:0000 entsize:0000 align:0001 => Program data
 [002] 0x08048128 a------ .note.ABI-tag   foffset:00000296 size:00000032 link:00
info:0000 entsize:0000 align:0004 => Notes
 [003] 0x08048148 a------ .hash           foffset:00000328 size:00000328 link:04
info:0000 entsize:0004 align:0004 => Symbol hash table
 [004] 0x08048290 a------ .dynsym         foffset:00000656 size:00000688 link:05
info:0001 entsize:0016 align:0004 => Dynamic linker symtab
 [005] 0x08048540 a------ .dynstr         foffset:00001344 size:00000430 link:00
info:0000 entsize:0000 align:0001 => String table
 ...
 [021] 0x0804CB58 aw----- .got            foffset:00015192 size:00000168 link:00
info:0000 entsize:0004 align:0004 => Program data
 [022] 0x0804CC00 aw----- .bss            foffset:00015360 size:00000364 link:00
info:0000 entsize:0000 align:0032 => BSS
 [023] 0x00000000 ------- .shstrtab       foffset:00015360 size:00000181 link:00
info:0000 entsize:0000 align:0001 => String table

 [*] Object /bin/cat unloaded
```

Si vous souhaitez plus de renseignements sur le format ELF, nous vous invitons à lire la documentation standard [elf].

Les conditions et problèmes rencontrés

Au tout début, nous nous sommes focalisés sur les binaires ELF à dissimuler, toujours en gardant à l'esprit qu'il faut distribuer l'information. Nous avons alors pensé à une solution un peu farfelue : on réduit la taille du binaire ELF au strict minimum (cf. [tinyelf]) et on le désassemble pour connaître ses instructions. On construit une liste qui contient le nom d'un fichier et un point de départ et de fin dans celui-ci. Sur la base de cette liste, on extrait et on agrège les instructions pour recréer le binaire. Chaque instruction de notre binaire est donc mappée avec des instructions équivalentes d'autres binaires.

Le principal avantage est qu'il n'y a rien d'écrit sur le système. En revanche, cette technique ne permet pas d'enregistrer un volume arbitraire de données, ce qui est un gros inconvénient. Il est inconcevable de récupérer les instructions assembleurs d'un binaire de 2Mo.

L'idée a alors été de dissimuler les données dans des sections d'autres binaires ou librairies ELF (binaires et librairies ayant la même structure ELF). Ces sections sont créées par nos soins et le problème d'espace de stockage ne se pose plus. Mieux encore, rendre ces sections invisibles dans les binaires grâce à la libelfsh

améliore le concept. C'est tout à fait faisable et pas compliqué du tout. Que fait la libelfsh en fait ? Elle modifie les pointeurs de la PHT et de la SHT de manière à ce que la section ne soit pas affichée lorsque l'on parcourt la SHT.

Reste maintenant à savoir comment découper le fichier à cacher. La première solution retenue est un découpage selon les sections (on cache donc des sections dans d'autres sections). Mais alors deux problèmes, auxquels nous n'avions pas pensé, surviennent.

Le premier vient de notre technique elle-même : il n'y a pas que des sections qui composent un binaire ELF, il y a aussi l'en-tête ELF, la PHT ou encore la SHT. L'en-tête ELF et la PHT se trouvent avant la première section et la SHT après la dernière section :

```
$ elfsh -f /bin/ls -e | grep "foffset"
 Data encoding      :      Little endian   SHT foffset        :       00071568
 PHT foffset        :           00000052   SHT entries number :             28
$ elfsh -f /bin/ls -s | tail -4
 [025] 0x00000000 ------- .shstrtab  foffset:00071362 size:00000204 link:00
info:0000 entsize:0000 align:0001 => String table

 [*] Object /bin/ls unloaded

$
```

On peut voir ici que la PHT se trouve à l'offset 52 et la SHT à l'offset 71568. La dernière section commence à l'offset 71362 et se termine à l'offset 71566 (dû à sa taille de 204 bytes). La SHT commence à l'offset 71568 dû à l'alignement sur 4 bytes. Avec la technique de découper en sections, tous ces en-têtes ne sont pas pris en compte. Il n'est cependant pas impossible d'arriver à les reconstruire à partir des sections. Cette technique aurait donc pu fonctionner avec une implémentation légèrement plus compliquée. Mais il y a surtout un autre problème qui, lui, est beaucoup plus contraignant.

Jusqu'à maintenant, nous nous sommes focalisés sur les binaires ELF, mais c'est une erreur. Il est nécessaire de pouvoir stocker tout type de données (ascii, binaire) de manière à pouvoir cacher des scripts (python, shell, etc.) ou des binaires. En découpant en section, on ne peut cacher que des binaires. C'est une erreur.

Pour remédier à cela, la solution retenue est de fractionner le fichier en blocs de données et d'injecter ces blocs dans des sections que nous ajoutons aux binaires. Un binaire ou fichier de données de 2M est divisé en 10 blocs de 200KB, chaque bloc étant placé dans une section dans un fichier ELF. On ne se soucie plus du type de fichier qu'on injecte.

Un détail, dans la première version du logiciel présentée par la suite, les sections ne sont pas créées n'importe où dans un fichier ELF. En principe, on peut le faire, mais il est beaucoup plus simple de la créer après la dernière section. Ceci pour une raison toute simple : si on crée une section après la dernière section, on devra modifier beaucoup moins de pointeurs que si on crée une section entre deux sections déjà existantes.

Avant insertion :

```
section_1

section_2

section_3
```

Après insertion :

```
section_1

section_evil

section_2

section_3
```

Si on insère une section entre la section 1 et 2, outre la taille du segment qui change, il y a aussi les sections 2 et 3 qui sont déplacées et donc leurs offsets sont modifiés. De plus, toutes les tables de relocations devront être recalculées. Ce n'est pas impossible à faire, mais c'est un travail titanesque et fort dépendant de l'architecture. Par simplicité, on insère les sections à la fin. La libelfsh insère automatiquement les sections à la fin (en réalité après la section .bss). Il est également possible d'insérer les sections par le bas, avant la première section. Mais ceci est utilisé surtout pour l'injection de code et non de données.

Comment récupérer les données cachées ?

La première idée pour récupérer les données cachées était d'utiliser un fichier de correspondance. Celui-ci référençait les sections et fichiers utilisés.

Voici à quoi il ressemblait :

```
10:10583:test1:rep/1:rep/10:rep/11:rep/12:rep/13:rep/14:rep/15:rep/16:rep/17:
rep/18
10:10583:test2:rep/19:rep/2:rep/20:rep/21:rep/22:rep/23:rep/24:rep/25:rep/26:
rep/27
[...]
10     : nombre de fichier ELF utilisés pour cacher le fichier
10583  : taille du fichier
test1  : nom du fichier
rep/1  : premier fichier ELF où est stocké le premier bloc
rep/10 : deuxième fichier ELF où est stocké le deuxième bloc
...
```

Chaque ligne correspond à un fichier caché. Cette solution était simple et fonctionnait très bien. Cependant, il fallait se « promener » avec le fichier de correspondance pour savoir quelles données étaient cachées sur une machine et à quel endroit elles l'étaient. Ce qui n'était pas très pratique. Nous avons alors opté pour la solution de stocker ces informations dans une section d'un fichier ELF (de la même manière que nous cachons les données). On utilise ainsi un fichier ELF de « configuration » dans lequel on stocke les lignes correspondant aux fichiers cachés. Il faut bien sûr utiliser toujours le même fichier de configuration sur un système, de manière à toujours se souvenir de celui à utiliser.

Ainsi, chaque fois que nous voulons cacher ou extraire un fichier, nous indiquons au programme quel fichier de configuration est à utiliser (toujours un binaire ou libraire ELF). En fonction de ce que nous voulons faire (cacher ou extraire des données), le programme va aller écrire ou lire dans le fichier de configuration. Il n'est plus nécessaire de se « promener » avec un fichier contenant les correspondances.

Nous pouvons pousser le vice plus loin en utilisant le noyau pour reconstruire notre binaire. On garde en mémoire kernel (ou ailleurs dans le kernel) ce tableau de correspondance. Lorsque nous voulons exécuter `/bin/superbinaire`, le noyau l'intercepte, et connaît les références pour reconstruire le superbinaire et

l'exécuter. Cette technique a été présentée pour la première fois par [palmers]. Malheureusement, avec cette technique, il faut garder en mémoire le tableau de correspondance. Cela ne pose pas de problème quand le système tourne. Mais dès qu'un redémarrage du système survient, la mémoire du noyau étant réinitialisée, le tableau n'est plus présent. Un autre désavantage avec cette technique est que l'on touche au noyau. Il faut y insérer un code à l'aide d'un module voire directement en passant par `/dev/kmem`. Comme c'est dit plus haut, si nous n'avons pas accès au noyau, c'est impossible. De plus, le noyau est souvent capricieux et instable. Sans parler des problèmes de compatibilité entre les différents noyaux (ex. : le noyau BSD est complètement différent du noyau Linux ou Solaris).

En bref, mieux vaut toujours utiliser un code dans l'espace utilisateur (ou *userland*) si le noyau n'est pas absolument nécessaire.

Le proof of concept

Le concept étant expliqué, nous allons montrer maintenant les grandes lignes de sa réalisation. N'espérez en aucun cas avoir, dans cet article, un outil complètement fonctionnel. Il permettra certes de cacher des données, mais il ne remplira pas toutes les conditions du cahier des charges que nous nous sommes fixés. Tout du moins la version que nous publierons. Le logiciel appelé DHIS pour *Distributed Hidden Storage* est toujours en développement et beaucoup de fonctionnalités doivent encore être intégrées.

Le cahier des charges

Avant de commencer à coder l'outil, il était nécessaire de mettre sur papier ses fonctionnalités. Outre le fait qu'il va distribuer les données à cacher dans les binaires ou librairies et qu'il va pouvoir les récupérer pour les réassembler, il devra pouvoir :

- Référencer les sections nouvellement créées selon leur position et leur taille. Dans le fichier de correspondance, on écrit seulement les fichiers ELF utilisés. Pour retrouver la section créée, on recherche le nom de la section dans la SHT. Mais, au moment de développer l'outil, un nouveau problème est apparu. Puisque, pour récupérer la section, on parcourt la SHT, si la section est cachée, elle ne sera plus présente dans la SHT, et donc impossible de la parcourir pour trouver la section qui nous intéresse. D'où l'intérêt de se baser sur leurs tailles et leurs offsets et non sur leurs noms.

- Choisir la manière de cacher les données. Nous souhaitons offrir la possibilité de choisir la méthode : soit par blocs injectés dans de nouvelles sections dans les fichiers, soit dans le padding entre segments (idéal pour cacher des fichiers de petites tailles) ou encore en utilisant des sections non mappées lors de l'exécution du binaire, comme la section `.comment`..

  Le fait d'utiliser des sections non mappées pour cacher des données va permettre, par rapport aux deux autres techniques, de laisser inchangé la taille du fichier ELF. La furtivité sera encore augmentée.

- Choisir les fichiers à infecter ou les sélectionner aléatoirement. Si on utilise la deuxième solution, il faut alors pouvoir détecter les fichiers déjà infectés pour éviter de les réutiliser. Nous n'avons pas retenu la solution d'ajouter plusieurs sections dans un même fichier.

- Chiffrer les sections que nous ajoutons dans les binaires. Un chiffrement symétrique (DES, AES ou *Blowfish*) pourra être utilisé. Ceci empêchera une analyse du binaire en brute, par exemple avec la commande `grep`.

- Permettre un système de réplication des données si un fichier ELF venait à disparaître. Ceci augmentant la taille des données à cacher, l'utilisateur aura la possibilité de choisir de répliquer ou pas. En attendant cette fonctionnalité, il peut très bien cacher deux fois le binaire s'il le désire.

- Avoir une fonction `wipe` qui efface correctement tout ce qui a pu être écrit sur le disque.

- Avoir la possibilité de ne rien écrire sur le disque si c'est un binaire ELF qui est reconstruit. Les données une fois `map()ées`, on reconstruit le binaire en mémoire et on l'exécute. Le problème, c'est le noyau qui gère l'exécution en appelant `sys_exec()` sur un nom de fichier et non sur une région mémoire `map()ée`. Intervient alors l'outil de The Grugq [ulexec] qui reproduit l'appel système `execve()` en mode *user*. The Grugq fournit une implémentation de son concept sous forme de librairie `libulexec.so`. En compilant notre binaire avec cette librairie, il sera capable une fois reconstruit et `map()é` en mémoire d'être exécuter sans aucune écriture sur le disque.

  Au moment de boucler cet article, une autre technique, [Pluf & Ripe], pour se passer de `execve()` a vu le jour dans le dernier Phrack.

- Pouvoir choisir le nombre de blocs de données pour la découpe du fichier. La taille des blocs sera aléatoire de manière à augmenter la discrétion.

Une ch'tite démo

DHIS est le nom donné à l'outil que vous trouverez à l'adresse [dhis]. Pour tout retour de bugs, de commentaires, etc. vous pouvez envoyer un mail à l'adresse **dhis@devhell.org**.

La démonstration qui suit est faite avec une version non finalisée, mais déjà fonctionnelle. Il se peut qu'au moment où vous lisez cet article des modifications aient été apportées.

```
# ./dhis
...
Usage   : [-h] [-l] [-v] [-c config_file] [-f file] [-d directory]
          [-b blocs] [-i file] [-e file] [-r file] [-m method]

Options :
                -h      display help
                -i      hide a file
                -l      list hidden files
                -e      extract hidden file
                -r      remove hidden file
                -d      directory to hide file
                -c      ELF file to use for storing config file (optionnal)
                -t      target file to save extracted file
                -v      verbose mode (to use with -l)
                -b      number of blocs to use (10 if no specified)
                -m      injection method:
                              0. append a section (by default)
                              1. use .comment section
```

```
                        2. use padding
Examples :
                ./dhis -i uname -d /usr/lib -c /usr/libm.so
                ./dhis -l -c /usr/libm.so
                ./dhis -e uname -d /usr/lib -c /usr/libm.so -t /tmp/uname
                ./dhis -r uname -c /usr/libm.so

#
```

Dans l'aide, nous retrouvons bien les différentes commandes qui permettent de cacher un fichier, de récupérer un fichier caché et de l'effacer si nous ne souhaitons plus le dissimuler. Pour l'exécuter, il suffira simplement de l'extraire, de lui rajouter le mode +x (exécution) si nécessaire et de le lancer. Dans cette configuration, on peut voir que le binaire est écrit sur le disque avant d'être exécuté. Ceci diminue la furtivité. Une amélioration, comme indiqué plus haut, sera d'utiliser un `execve userland` permettant de ne pas devoir écrire le binaire sur le disque, mais de directement l'exécuter en mémoire.

```
# ./test1
Test 1!
# ls misc/
cat  cp  date  dmesg  ln  ls  mkdir  mv  pwd  touch
# ./dhis -i test1 -d misc/ -c misc/cat
...
[+] injection: append a section to binary
[+] hide file test1 in directory misc/
[+] some checks before to hide your file
  [+] ok, we've got enough not infected binaries in misc/
  [+] config file misc/cat is a ELF file
[+] start to hide file test1
  [+] append section .dhis to misc/mv
  [+] append section .dhis to misc/pwd
  [+] append section .dhis to misc/mkdir
  [+] append section .dhis to misc/ls
  [+] append section .dhis to misc/ln
  [+] append section .dhis to misc/dmesg
  [+] append section .dhis to misc/date
  [+] append section .dhis to misc/cp
  [+] append section .dhis to misc/touch
  [+] 10:11134:test1:misc/mv:misc/pwd:misc/mkdir:misc/ls:misc/ln:misc/dmesg:
misc/date:misc/cp:misc/touch
[+] file test1 hidden
  [+] append section .dhis to misc/cat
[+] config hidden in file misc/cat

#
```

Que s'est-il passé ? Nous avons dissimulé le fichier `test1` dans des binaires ELF situés dans le répertoire `misc`. 10 binaires ont eu la section `.dhis` ajoutée dans laquelle un bloc de `test1` a été copié (la commande `-s` de elfsh permet de lister les sections) :

```
$ elfsh -f misc/mv -s
[...]
 [023] 0x0804CD6C a-x---- .dhis                          foffset:00015724
size:00001058 link:00 info:0000 entsize:0000 align:0000 => Program data
[...]
$ elfsh -f misc/pwd -s
[...]
 [024] 0x0805554C a-x---- .dhis                          foffset:00050496
size:00001058 link:00 info:0000 entsize:0000 align:0000 => Program data
[...]
```

Et dans le fichier de correspondance (cf. « Comment récupérer les données cachées ? »), ici `misc/cat`, nécessaire pour la reconstruction du fichier (la commande `-X` de elfsh permet d'afficher le contenu d'une section en hexadécimal) :

```
# elfsh -f misc/cat -X .dhis

 [*] Object misc/cat has been loaded (O_RDONLY)

 0804C90C [foff: 00014604] .dhis + 0                     31 30 3A 31 31 31 33
34 3A 74 65 73 74 31 3A 6D 10:11134:test1:m
 0804C91C [foff: 00014620] .dhis + 16                    69 73 63 2F 6D 76 3A
6D 69 73 63 2F 70 77 64 3A isc/mv:misc/pwd:
 0804C92C [foff: 00014636] .dhis + 32                    6D 69 73 63 2F 6D 6B
64 69 72 3A 6D 69 73 63 2F misc/mkdir:misc/
 0804C93C [foff: 00014652] .dhis + 48                    6C 73 3A 6D 69 73 63
2F 6C 6E 3A 6D 69 73 63 2F ls:misc/ln:misc/
 0804C94C [foff: 00014668] .dhis + 64                    64 6D 65 73 67 3A 6D
69 73 63 2F 64 61 74 65 3A dmesg:misc/date:
 0804C95C [foff: 00014684] .dhis + 80                    6D 69 73 63 2F 63 70
3A 6D 69 73 63 2F 74 6F 75 misc/cp:misc/tou
 0804C96C [foff: 00014700] .dhis + 96                    63 68
                           ch

 [*] Object misc/cat unloaded

#
```

La commande `-l` de dhis permet de lister les fichiers cachés :

```
# ./dhis -l -c misc/cat
...
[+] list hidden files

NUM  BLOC    SIZE      NAME
1    10      11134     test1

#
```

La taille indiquée (11134) est la taille totale du binaire. Chaque bloc inséré est de taille différente. Parfois, la taille du bloc injecté est de 700 bytes, parfois 1000 bytes, parfois 1200 bytes,... La furtivité est ainsi augmentée. Concernant le nom de la section, il est toujours le même pour le moment, mais celui-ci sera aléatoire par la suite. On peut se le permettre puisqu'on se basera sur l'offset et la taille pour retrouver une section et non son nom.

Si on édite la section `.dhis` du premier binaire infecté, on peut noter la présence de l'en-tête ELF du binaire `test1` à l'offset 0 :

```
# elfsh -f misc/mv -X .dhis | more

[*] Object misc/mv has been loaded (O_RDONLY)

0804CD6C [foff: 00015724] .dhis + 0                     7F 45 4C 46 01 01 01
00 00 00 00 00 00 00 00 00 .ELF.............
0804CD7C [foff: 00015740] .dhis + 16                    02 00 03 00 01 00 00
00 C0 82 04 08 34 00 00 00 .............4...
[...]
```

Il est temps maintenant de reconstruire le binaire dissimulé de manière à pouvoir l'exécuter :

```
# ./dhis -e test1 -d misc/ -c misc/cat -t /tmp/newfile
[+] extract hidden file test1
  [+] get section .dhis in misc/mv
  [+] get section .dhis in misc/pwd
  [+] get section .dhis in misc/mkdir
  [+] get section .dhis in misc/ls
  [+] get section .dhis in misc/ln
  [+] get section .dhis in misc/dmesg
  [+] get section .dhis in misc/date
  [+] get section .dhis in misc/cp
  [+] get section .dhis in misc/touch
[+] file saved under name /tmp/newfile

# chmod +x /tmp/newfile
# /tmp/newfile
Test 1!
#
```

Il suffit d'indiquer au programme le répertoire utilisé ainsi que le fichier de configuration. Un nom de fichier de destination sera aussi nécessaire, ici nous avons choisi `/tmp/newfile`. Il suffira ensuite simplement de changer le mode du fichier en exécution et de l'exécuter.

Imaginez maintenant que nous souhaitons cacher un second fichier :

```
# ./dhis -i test2 -d misc/ -c misc/cat
[+] injection: append a section to binary
[+] hide file test2 in directory misc/
[+] some checks before to hide your file
  [+] no enough file, all files are already infected
#
```

Il détecte les fichiers déjà infectés.

Beaucoup de fonctionnalités sont encore à ajouter dans `dhis`, notamment pour rendre la technique d'injection la plus discrète possible. Par exemple, au moment de boucler cet article, le logiciel n'était pas encore capable de cacher une section et de la retrouver par la suite. Malgré tout, la libelfsh propose cette fonctionnalité et l'implémenter dans le logiciel n'est qu'une question de temps. Comme expliquer plus haut, les pointeurs de la SHT et de la PHT auront été modifiés pour qu'elle ne soit pas présente. Au moment où vous lirez ces lignes, la fonction sera plus que probablement déjà incluse dans le logiciel.

La solution de détection

Pour détecter les données injectées, il est assez facile de vérifier la taille des sections et leurs adresses de départ et de fin. Le calcul est simple : si l'adresse de fin de la `section_1` (= adresse de départ+taille de la section) n'est pas égale à l'adresse de départ de la `section_2 - 1`, alors il y a des données cachées entre la `section_1` et la `section_2`. Cependant, comme on peut facilement cacher les sections, cette technique s'avère inutile. Le mieux est de comparer la taille de chaque segment avec la somme des sections appartenant à ce segment. Si c'est différent, alors il y a une section cachée dans ce segment. Notons tout de même que les deux tailles seront toujours différentes à cause du padding, mais si une section est ajoutée, la différence sera, en général, plus grande que 4 k.

La solution la plus simple reste cependant d'utiliser un logiciel comme [tripwire]. Quand le système est sain, tripwire sauvegarde dans un fichier l'état du *filesystem*. Il sauvegarde entre autres la taille des fichiers, leurs dates de création ainsi qu'un *hash* des fichiers. Si le système est compromis, il est ensuite facile de comparer l'état des fichiers avec le fichier de sauvegarde pour détecter les fichiers altérés.

Conclusion

La technique présentée ici n'a rien de révolutionnaire. Injecter des données dans des fichiers se fait depuis de nombreuses années sur de nombreux systèmes d'exploitation. Par contre, le fait que ce soit distribué apporte un plus considérable au niveau de la furtivité par rapport aux autres types d'injections.

Quand un pirate arrive sur une machine, après avoir obtenu l'accès administrateur ou *root*, une de ces premières tâches sera de garder son accès. Une *backdoor* sera alors installée, celle-ci sera choisie en fonction de la configuration du système (*backdoor kernelland* ou *userland*). Dans de nombreux cas, il voudra également stocker des fichiers sur la machine. Cela pourra être des logiciels d'analyses réseau, des exploits ou même des fichiers confidentiels qu'il voudra mettre en lieu sûr. S'il veut toucher au kernel, il pourra utiliser un code kernel pour cacher ses fichiers. Dans le cas contraire, il devra cacher ses fichiers en userland et c'est là que notre logiciel DHIS interviendra.

Bibliographie

[dhis] Programme développé pour cet article ; **http://dhis.devhell.org**

[elf] TIS Consortium. *Elf portable format specifications*, TIS, 1998

[elfsh] Elfsh Team – *The ELFsh project* ; **http://elfsh.devhell.org**

[tinyelf] **http://www.muppetlabs.com/~breadbox/software/tiny/teensy.html**

[plaguez] plaquez – *Weakening the Linux Kernel* ; **http://www.phrack.org/show.php?p=52&a=18**

[runefs] The Grugq – *Defeating Forensic Analysis on Unix* ; **http://www.phrack.org/phrack/59/p59-0x06.txt**

[altplt] mayhem – *The Cerberus ELF Interface* ; **http://www.phrack.org/phrack/61/p61-0x08_The_Cerberus_ELF_interface.txt**

[elfparasite] Silvio Cesare – *Unix ELF parasites and virus* ; **http://www.uebi.net/silvio/elf-pv.txt**

[vit] Silvio Cesare – Virus VIT ; **http://www.uebi.net/silvio/vit.html**

[siilov] Silvio Cesare – Virus SIILOV ; **http://www.uebi.net/silvio/siilov.txt**

[palmers] palmers – *Advances in kernel hacking II* ; **http://www.phrack.org/show.php?p=59&a=5**

[ulexec] The Grugq – *The Design and Implementation of Userland Exec* ;
http://lists.grok.org.uk/pipermail/full-disclosure/attachments/20040101/fea4fb1f/ul_exec.txt

[Pluf & Ripe] Pluf & Ripe – *Advanced antiforensics : SELF* ; **http://www.phrack.org/show.php?p=63&a=11**

[tripwire] **http://sourceforge.net/projects/tripwire/**

Le reverse engineering facile avec DTrace

L'une des difficultés rencontrées lors de l'analyse du fonctionnement d'un programme binaire est de découvrir la portion de code réalisant une action précise. Cet article montre comment découvrir la fonction responsable du hachage des mots de passe Oracle en utilisant DTrace.

Introduction à DTrace

DTrace est un système complet d'analyse dynamique d'un système sous Solaris 10. Il est principalement utilisé par les développeurs et administrateurs pour obtenir des données pertinentes concernant le comportement du système d'exploitation ou des processus. Ses principales qualités sont sa souplesse et sa stabilité.

Pour ne rien gâcher, les développeurs responsables de cet outil sont très sympathiques et répondront à vos questions les plus tordues. Ne soyez pas de ceux qui laissent DTrace au plafond !

DTrace fonctionne en modifiant dynamiquement le système d'exploitation ou les processus utilisateurs de sorte à enregistrer des informations en des points appelés « probes ». A chaque probe est associé un ensemble d'« actions », définies par l'utilisateur.

Il est très fortement recommandé de se procurer l'excellente documentation officielle [1] ainsi que de consulter les blogs des développeurs DTrace [2, 3, 4], bourrés d'exemples et de nouveautés.

Le langage D

Un langage de script, appelé « le langage D », est utilisé pour définir les probes et actions.

Sans entrer dans les détails, toutes les probes utilisées dans cet article font partie du *provider* `pid`, un provider étant un module noyau chargé d'instrumenter une catégorie particulière d'événements. Le provider `pid` est spécialisé dans l'analyse des processus utilisateurs.

Voici un exemple de script D :

```
#!/usr/sbin/strace
pid$target::doubler:entry
/arg0>0/
{
        trace(arg0);
}
pid$target::doubler:return
/arg1!=4/
{
        trace(arg1);
}
```

Que se passe-t-il ? La structure d'un script D est presque toujours composée d'une liste de probes, prédicats et actions :

Les probes

Exemple :

```
pid$target:module:doubler:entry
```

Elle s'analyse ainsi :

- `pid$target` est le provider. Ici on désigne le provider **pid**, associé au processus **$target**, variable qui désigne « automagiquement » le processus exécuté au moyen de l'option `-c` de DTrace.
- `module` désigne ici la bibliothèque à observer, c'est-à-dire par exemple `libc.so.1`. A noter qu'il est possible d'utiliser l'alias `a.out` pour désigner le binaire lui-même.
- `doubler` est le nom de la fonction à observer.
- `entry` est la position de la probe, ici à l'entrée de la fonction. Il est possible de désigner une position arbitraire dans une fonction, mais seules seront utilisées `entry` et `return` dans cet article.

Les prédicats

Les prédicats sont des conditions logiques qui permettent de décider si oui ou non les actions seront exécutées lorsqu'une probe est atteinte. Ici on a par exemple :

```
/arg0>0/
```

Les variables `argN` contiennent les valeurs des arguments passés à la fonction. Si la position de la probe est return, `arg1` contient la valeur de retour de la fonction.

Ici, le prédicat vérifie que le premier argument passé à la fonction est positif non nul avant d'exécuter les actions.

Les actions

Les actions sont les opérations qui sont exécutées lorsqu'une probe est atteinte est que le prédicat est vérifié. La commande `trace` permet d'afficher des valeurs. Dans cet article seront utilisés `printf`, qui et presque identique à la fonction C du même nom, `tracemem` qui permet d'afficher une portion de mémoire et `ustack` qui affiche la trace d'exécution de la pile.

Magie !

Le script précédent est utilisé ainsi :

```
$ dtrace -s test.d -c ./test
```

Résultat :

```
CPU     ID                    FUNCTION:NAME
  0  34947                     doubler:entry        0
  0  34948                    doubler:return        0
  0  34947                     doubler:entry        1
  0  34948                    doubler:return        2
  0  34947                     doubler:entry        2
```

Simon Marechal
simon.marechal@thales-security.com

```
0  34948              doubler:return          4
0  34947               doubler:entry          3
0  34948              doubler:return          6
0  34947               doubler:entry          4
0  34948              doubler:return          8
```

Que lire ? L'affichage est organisé sous forme de colonnes. La première représentant le CPU sur lequel le code examiné s'exécute (toujours 0 pour les systèmes uni-processeurs). La colonne ID représente l'identifiant de la probe utilisée. Cet identifiant est utilisé en interne par DTrace et n'est pas bien utile. Finalement sont affichées les informations utiles : la fonction examinée, la position de la probe dans cette fonction et les données tracées.

Mise en pratique

Hypothèse

L'hypothèse au départ de cette recherche est que le client oracle `sqlplus` s'authentifie à la base de données en utilisant une forme de challenge-réponse. Ceci signifie, entre autres, qu'il est capable de calculer le hachage du mot de passe utilisateur.

Le couple utilisateur/mot de passe utilisé dans cet article est `SYSTEM/THALES`. Le hachage correspondant étant `9EEDFA0AD26C6D52`.

Recherche d'une fonction retournant le mot de passe

La première idée est de rechercher une fonction retournant notre mot de passe. Une première approche naïve est de faire ainsi :

```
#!/usr/sbin/dtrace -s
pid$target:::return
/(arg1 == 0x9EEDFA0A) || (arg1 == 0x0AFAED9E)/
{
        trace(probemod);
}
```

C'est-à-dire rechercher les fonctions retournant la première partie du hachage. En effet, celui-ci est trop grand pour tenir dans un registre 32bit. On remarque que le test est effectué avec les versions grand et petit endien du hachage. Oracle étant en effet multiplateforme, la représentation interne du mot de passe est inconnue. Ce test n'est pas concluant, il faut trouver mieux.

Une approche plus pertinente est de rechercher les fonctions qui retournent un pointeur vers le hachage.

```
#!/usr/sbin/dtrace -s

this int * p;

pid$target:::return
/(arg1 > 0x8000000) && (arg1<0x8100000)/
{
        this->p = copyin(arg1, 8);
        printf("%8x%8x", this->p[0], this->p[1]);
}
```

Un casseur de mot de passe oracle ?

Un message [5] a été posté contenant l'algorithme de hachage étudié durant la rédaction de cet article. A la suite, de nombreux outils jusque-là publics ont été dévoilés [6]. Les plus efficaces sont :

- Orabf : le plus rapide, tout simplement. Sa vitesse très importante chute dès lors qu'une attaque par dictionnaire est utilisée au lieu de la force brute.
- Le plugin pour John the Ripper [7] : pas le plus rapide en vitesse pure, il compense ce défaut par la qualité du générateur de mot de passe, la disponibilité des sources et le fait qu'il soit multiplateforme.

Énumération des comptes utilisateurs

Il est possible de déterminer si un compte utilisateur est valide ou non en regardant si le mot de passe est haché ou pas. Une analyse approfondie du protocole de communication permettrait d'écrire un énumérateur de comptes générique. Avis aux amateurs !

Ce programme présente certaines particularités :

- Le mot clef `this` permet de définir une variable qui est locale à un *thread*, de sorte à éviter les accès concurrents sur les processus *multithreadés*.
- Le prédicat permet de vérifier (d'une manière complètement non scientifique) que l'argument retourné par une fonction est bien un pointeur vers une zone mémoire valide.
- La fonction `copyin` permet de copier des données provenant de l'espace utilisateur vers l'espace noyau sur lequel DTrace peut travailler. Elle prend en argument un pointeur et une taille. On remarquera que le pointeur est invalide, un simple message d'erreur est retourné, mais que l'exécution continue.

Le résultat est, après un filtrage approprié :

```
0  59252        memcpy:return  afaed9e526d6cd2
```

Ce résultat est notre mot de passe, après conversion d'endianité !

La piste du memcpy

Suivre la trace...

En observant les arguments de cette fonction `memcpy`, il apparaît que l'adresse mémoire destination est fixe, c'est-à-dire que le mot de passe est toujours inscrit dans la même zone mémoire.

Il est donc possible de tracer l'utilisation de cette zone avec le script suivant :

```
pid$target::memcpy:entry
/arg0 == 0x8043ec0/
{
        self->trace = 1;
}

pid$target:::entry,pid$target:::return
/self->trace/
{
        printf("%8x %8x %8x %8x", arg0, arg1, arg2, arg3);
}
```

Ce script a pour but d'afficher des détails sur tous les appels de fonction à partir du moment où ce tampon est utilisé par la fonction memcpy.

Un extrait de la sortie est :

```
CPU     ID                    FUNCTION:NAME
  0  35271          memcpy:entry 8043ec0  8043fdc       40 8043fc0
  0  87542           memcpy:return     1de  8043ec0       40 0
  0  69427        ztchsh1h:return    1097 2b6ee1e4 668ae164 0
  0  41675         ztchsh1h:entry 8043fc0  8044080  80445a4 8043fc0
  0  35271            0  35271      memcpy:entry 8043ec0  8043fdc       40 8043fc0
  0  87542           memcpy:return     1de  8043ec0       40 0
  0  69427        ztchsh1h:return    1097 f9536936 d9df074b 0
  0  41675         ztchsh1h:entry 8043fc0  80440c0  80445a4 8043fc0
  0  35271           memcpy:entry 8043ec0  80440c0       40 8043fc0
  0  87542           memcpy:return     1de  8043ec0       40 0
  0  69427        ztchsh1h:return    1097 f29332ab 8196bafe 0
  0  41675         ztchsh1h:entry 8043fc0  8044100  80445a4 8043fc0
  0  35271           memcpy:entry 8043ec0  8044100       40 8043fc0
  0  87542           memcpy:return     1de  8043ec0       40 0
  0  69427        ztchsh1h:return    1097 4d891880 4e383ec4 0
  0  41675         ztchsh1h:entry 8043fc0  8044140  80445a4 8043fc0
  0  35271           memcpy:entry 8043ec0  8044140       40 8043fc0
  0  87542           memcpy:return     1de  8043ec0       40 0
  0  69427        ztchsh1h:return    1097 fe10ba21   a86c2f 0
  0  41675         ztchsh1h:entry 8043fc0  8044180  80445a4 8043fc0
  0  35271           memcpy:entry 8043ec0  8044180       40 8043fc0
...
```

Le tampon est utilisé à plusieurs reprises par la fonction memcpy. On remarque également que cette fonction memcpy est celle utilisée dans la fonction ztchsh1h. Il est également possible d'observer que le second argument de ztchsh1h est également le second argument de la fonction memcpy.

...et c'est l'échec

Que déduire de ces informations ? Et quelle est cette fonction ztchsh1h ? Tout d'abord examinons les arguments des appels à la fonction memcpy :

```
0  35271      memcpy:entry 8043ec0  8043fdc       40
0  35271      memcpy:entry 8043ec0  80440c0       40
0  35271      memcpy:entry 8043ec0  8044100       40
```

La fonction memcpy, comme son nom l'indique, est utilisée pour copier des zones mémoire. Le premier argument, qui est un pointeur vers une zone mémoire fixe dans laquelle passe notre mot de passe est l'argument destination. C'est également la valeur de retour de cette fonction.

Le second argument est l'adresse source. On remarque que cette adresse est incrémentée de 64 octets entre chaque appel à memcpy. Le troisième paramètre est la taille de la zone mémoire à copier, également 64 octets (0x40 en hexadécimal). Une zone mémoire d'une taille plus importante est donc copiée vers un tampon fixe par blocs de 64 octets.

Qu'en est-il de la fonction ztchsh1h ? Un coup d'œil aux fonctions présentant un nom similaire au moyen de la commande objdump donne le résultat suivant :

```
$ objdump -t libclntsh.so  | grep ztch
...
0034dd40 l     F .text  00001098 ztchsh1h
0034eefe g     F .text  000000f5 ztchsh1f
0034ede0 g     F .text  00000034 ztchsh1i
0034ee21 g     F .text  000000c0 ztchsh1n
0034cca1 g     F .text  00000080 ztchmd5f
0034d490 g     F .text  0000002d ztchmd5i
0034d4c5 g     F .text  000000be ztchmd5n
0034d691 g     F .text  0000008d ztchmd4f
0034d590 g     F .text  0000002d ztchmd4i
0034d5c4 g     F .text  000000bd ztchmd4n
...
```

On observe le nom de fonctions de hachage (MD5, MD4). On peut en déduire sans trop de risques que la fonction ztchsh1h est liée à la fonction de hachage SHA1. De plus, ces fonctions sont généralement implémentées ainsi : une fonction permet d'initialiser le hachage (*Init*), une fonction permet de passer les données à hacher (*Update*), une fonction est utilisée en interne pour réaliser les calculs intermédiaires (*Body*) et une fonction retourne le hachage (*Final*).

Il y a donc fort à parier que ztchsh1i soit la fonction *Init*, ztchsh1n *Update*, ztchsh1f *Final* et donc que ztchsh1h soit la fonction *Body* ! Ceci cadre assez bien avec les observations précédentes, la fonction *Body* de SHA1 calculant les hachages par blocs de 64 octets.

Malheureusement, les observations indiquent que le mot de passe est haché par SHA1, et non pas le résultat de cette fonction. Le hachage doit donc être calculé en amont, retour à la case départ...

Recherche d'une fonction qui prend le mot de passe en argument

Comme précédemment, une fonction dont l'un des arguments est un pointeur vers le mot de passe a été recherchée :

```
#!/usr/sbin/dtrace -s
this int * p;
pid$target:::entry
/(arg0 > 0x8000000) && (arg0 < 0x8100000)/
{
        this->p = copyin(arg0, 8);
        printf("%s %8x%8x", probemod, this->p[0], this->p[1]);
}
```

La variable probemod contient le nom de la bibliothèque dans laquelle se trouve la fonction recherchée. En filtrant la sortie on découvre :

```
  0  35095          enter:entry LM1`ld.so.1 9eedfa0ad26c6d52
  0  41431        ztuc8tx:entry libclntsh.so.10.1 9eedfa0ad26c6d52
```

Ces deux fonctions prennent en premier argument un pointeur vers le mot de passe. La première faisant partie de la bibliothèque système ld, elle ne sera pas étudiée. La seconde est plus intéressante.

Pour découvrir d'où vient le mot de passe, la fonction ustack est utilisée pour découvrir la fonction qui a passé le mot de passe :

```
#!/usr/sbin/dtrace -s
pid$target::ztuc8tx:entry
{
        ustack();
}
```

Cette fonction affiche la trace d'exécution de la pile :

```
CPU     ID                    FUNCTION:NAME
  0  34947                    ztuc8tx:entry
              libclntsh.so.10.1`ztuc8tx
              libclntsh.so.10.1`ztvovgn+0xb3
              libclntsh.so.10.1`ztvovg+0x9e
              libclntsh.so.10.1`ztv2gorcl+0x68
              libclntsh.so.10.1`kzsr5gvfr+0x37f
              ...
```

La fonction `ztuc8tx` est donc appelée par `ztvovgn`. En observant cette dernière fonction, il est possible de remarquer que ses arguments sont :

- Un pointeur vers un tampon qui sera utilisé pour stocker le mot de passe haché, en représentation ASCII ;
- Un pointeur vers le nom de l'utilisateur ;
- La taille du nom de l'utilisateur ;
- Un pointeur vers le mot de passe ;
- La taille du mot de passe.

Le script suivant le démontre :

```
#!/usr/sbin/dtrace -s
this int i;

pid$target::ztvovgn:entry
{
        printf("%8x %s %s", arg0, stringof(copyin(arg1, arg2)),
stringof(copyin(arg3, arg4)));
        this->i = arg0;
}

pid$target::ztvovgn:return
{
        printf("%s", stringof(copyin(this->i,16)));
}
```

```
CPU     ID                    FUNCTION:NAME
  0  34947                    ztvovgn:entry  8043c08 thales SYSTEM
  0  34948                   ztvovgn:return 9EEDFA0AD26C6D52
```

On remarquera que la valeur de `arg0` est stockée dans une variable. En effet, lorsque la probe « `return` » est atteinte, arg0 contient l'adresse relative du pointeur d'exécution par rapport au début de la fonction. Pour réaliser un calculateur de hachage, il suffirait donc d'utiliser cette fonction. Elle n'est malheureusement pas exportée et ne peut donc pas être directement utilisée par un programme.

Analyse de la fonction ztvovgn

Affichage d'un flot d'exécution

La première étape pour analyser le fonctionnement de la fonction `ztvovgn` est d'analyser son flot d'exécution :

```
#!/usr/sbin/dtrace -s
#pragma D option flowindent
pid$target::ztvovgn:entry
{
        self->trace = 1;
}

pid$target:::entry
/(self->trace) &&
  (probemod != "libc.so.1") && (probemod != "LM1`ld.so.1")/
{
        printf("%8x %8x %8x %s", arg0, arg1, arg2, probemod);
}

pid$target:::return
/(self->trace) &&
  (probemod != "libc.so.1") && (probemod != "LM1`ld.so.1")/
{
        printf("%8x                %s", arg1, probemod);
}

pid$target::ztvovgn:return
{
      self->trace = 0;
}
```

On remarquera l'utilisation de l'option `flowindent`, qui permet de créer un flot indenté, et les prédicats qui permettent d'éviter de tracer tous les appels aux fonctions contenues dans la `libc` et la `libld` :

```
CPU FUNCTION
  0  -> ztvovgn           8043c18  8043c7c        6 libclntsh.so.10.1
  0    -> ztvovg_xmute    80438c0  8043b30  8043c7c libclntsh.so.10.1
  0      -> lxsNormStr    8043690     1f0  8043c5c libclntsh.so.10.1
  0      <- lxsNormStr          c                  libclntsh.so.10.1
  0      -> lxsNormStr    804369c     1e4  8043c7c libclntsh.so.10.1
  0      <- lxsNormStr          c                  libclntsh.so.10.1
  0    <- ztvovg_xmute          0                  libclntsh.so.10.1
  0    -> ztcedchk       d25de878  80438c0        6 libclntsh.so.10.1
  0      -> ztcedecb     d25de878  8043888  8043888 libclntsh.so.10.1
  0      <- ztcedecb     c5fea9b4                  libclntsh.so.10.1
  0      -> ztcedecb     d25de878  8043888  8043888 libclntsh.so.10.1
  0      <- ztcedecb     fb708a97                  libclntsh.so.10.1
  0      -> ztcedecb     d25de878  8043888  8043888 libclntsh.so.10.1
  0      <- ztcedecb     1de0e883                  libclntsh.so.10.1
  0    <- ztcedchk       1de0e883                  libclntsh.so.10.1
  0    -> ztcedgks        8043b3c  8043ab0        1 libclntsh.so.10.1
  0      -> ztced_einit   8043ab0  8043b3c d25aa09c libclntsh.so.10.1
  0      <- ztced_einit   816e6c7                  libclntsh.so.10.1
  0    <- ztcedgks        816e6c7                  libclntsh.so.10.1
  0    -> ztcedchk        8043ab0  80438c0        6 libclntsh.so.10.1
  0      -> ztcedecb      8043ab0  8043888  8043888 libclntsh.so.10.1
  0      <- ztcedecb      5617d0fb                  libclntsh.so.10.1
  0      -> ztcedecb      8043ab0  8043888  8043888 libclntsh.so.10.1
  0      <- ztcedecb      a1d00f60                  libclntsh.so.10.1
  0      -> ztcedecb      8043ab0  8043888  8043888 libclntsh.so.10.1
  0      <- ztcedecb      d26c6d52                  libclntsh.so.10.1
  0    <- ztcedchk        d26c6d52                  libclntsh.so.10.1
  0    -> ztuc8tx         8043b44  8043c18  8043b80 libclntsh.so.10.1
  0      -> ztuc4tx      9eedfa0a  8043c18  80438c0 libclntsh.so.10.1
  0      <- ztuc4tx            39                  libclntsh.so.10.1
  0      -> ztuc4tx      d26c6d52  8043c20  80438c0 libclntsh.so.10.1
  0      <- ztuc4tx            44                  libclntsh.so.10.1
  0    <- ztuc8tx              44                  libclntsh.so.10.1
  0  <- ztvovgn                0                  libclntsh.so.10.1
```

Analyse de la fonction ztcedchk

Observation des arguments

Cette fonction est appelée deux fois. Un script très générique permet d'obtenir de nombreuses informations sur la façon dont elle est utilisée :

```
this int * p;
pid$target::ztcedchk:entry
{
    printf("\nargs  : %8x %8x %8x %8x %8x\n", arg0, arg1, arg2, arg3, arg4);
    this->p = copyin(arg0,24);
    printf("inarg0: %8x %8x %8x %8x %8x %8x\n", this->p[0], this->p[1], this-
>p[2], this->p[3], this->p[4], this->p[5]);
    this->p = copyin(arg1,24);
    printf("inarg1: %8x %8x %8x %8x %8x %8x\n", this->p[0], this->p[1], this-
>p[2], this->p[3], this->p[4], this->p[5]);
    this->p = copyin(arg3,24);
    printf("inarg3: %8x %8x %8x %8x %8x %8x\n", this->p[0], this->p[1], this-
>p[2], this->p[3], this->p[4], this->p[5]);
    this->p = copyin(arg4,24);
    printf("inarg4: %8x %8x %8x %8x %8x %8x\n", this->p[0], this->p[1], this-
>p[2], this->p[3], this->p[4], this->p[5]);
}

pid$target::ztcedchk:return
{
        printf("return: %8x", arg1);
}
```

Le résultat :

```
CPU     ID                    FUNCTION:NAME
  0  34947                    ztcedchk:entry
args  : d25de878  80438b0        6  8043b24  8043b2c
inarg0: 19192410 90b39032  5242616 91a26161 312d1222 210142e1
inarg1:   530059   530054   45004d   540048   41004c   450053
inarg3:        0        0        0        0  8043bd8 d272013c
inarg4:        0        0  8043bd8 d272013c  8043bd8 d1f520b1
  0  34948                ztcedchk:return return: 1de0e883
  0  34947                ztcedchk:entry
args  :  8043aa0  80438b0        6  8043b24  8043b34
inarg0: 1c310013  3e3e2e0 25303633 712150a2 210e1c0f a2f062d0
inarg1:   530059   530054   45004d   540048   41004c   450053
inarg3:        0        0 dcd2472d 1de0e883  8043bd8 d272013c
inarg4:  8043bd8 d272013c  8043bd8 d1f520b1  8043c08  8043c6c
  0  34948                ztcedchk:return return: d26c6d52
```

C'est le contenu de l'argument 1 qui frappe l'œil ici, car il semble contenir une chaîne unicode. La fonction `tracemem` permet de s'en assurer :

```
pid$target::ztcedchk:entry
{
          tracemem(copyin(arg1,64), 64);
}
```

```
CPU     ID                    FUNCTION:NAME
  0  34947                    ztcedchk:entry
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  0123456789abcdef
 0: 59 00 53 00 54 00 53 00 4d 00 45 00 48 00 54 00  Y.S.T.S.M.E.H.T.
10: 4c 00 41 00 53 00 45 00 30 00 00 00 e0 11 a2 d1  L.A.S.E.0.......
20: d0 f6 19 08 00 00 00 00 60 aa d1 08 0a 00 00    ........`......
30: f8 38 04 08 29 0e a2 d1 d0 74 aa d1 00 00 00 00  .8..)....t......

  0  34947                    ztcedchk:entry
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  0123456789abcdef
 0: 59 00 53 00 54 00 53 00 4d 00 45 00 48 00 54 00  Y.S.T.S.M.E.H.T.
10: 4c 00 41 00 53 00 45 00 30 00 00 00 e0 11 a2 d1  L.A.S.E.0.......
20: d0 f6 19 08 00 00 00 00 60 aa d1 08 0a 00 00    ........`......
30: f8 38 04 08 29 0e a2 d1 d0 74 aa d1 00 00 00 00  .8..)....t......
```

On trouve ici une chaîne unicode contenant le nom d'utilisateur suivi du mot de passe, mais après une conversion d'endianité. Il aurait été plus aisé de l'afficher en utilisant le format `%ws` de la fonction `printf`, utilisé pour afficher des « *wide char strings* », mais cette fonctionnalité semble avoir été oubliée de DTrace…

Un coup d'œil sur la fonction ztcedecb

Un petit coup d'`objdump` permet de trouver ça :

```
2f88a2:       81 e2 33 33 33 33       and    $0x33333333,%edx
2f88a8:       33 ca                   xor    %edx,%ecx
2f88aa:       c1 e2 02                shl    $0x2,%edx
2f88ad:       33 d0                   xor    %eax,%edx
2f88af:       8b c1                   mov    %ecx,%eax
2f88b1:       c1 e8 08                shr    $0x8,%eax
2f88b4:       33 c2                   xor    %edx,%eax
2f88b6:       25 ff 00 ff 00          and    $0xff00ff,%eax
2f88bb:       33 d0                   xor    %eax,%edx
2f88bd:       c1 e0 08                shl    $0x8,%eax
2f88c0:       33 c8                   xor    %eax,%ecx
2f88c2:       8b c2                   mov    %edx,%eax
2f88c4:       d1 e8                   shr    %eax
2f88c6:       33 c1                   xor    %ecx,%eax
```

L'œil exercé remarquera que ça ressemble à du DES (précisément l'implémentation optimisée 32 bits).

Élémentaire mon cher Watson !

A cette étape, il est possible de réaliser quelques hypothèses :

- `ztcedecb` chiffre un bloc de 8 octets en DES.
- `ztcedchk` est un « mode » DES (tel que CBC).
- `ztcedgks` initialise le *key schedule*, c'est-à-dire qu'il accepte en argument une clef de 8 octets utilisée par DES.
- `ztvovg_xmute` appelle deux fois `lxsNormStr`. Avec un tel nom, il est probable qu'il s'agisse d'une fonction qui mette des chaînes de caractères sous une forme « normale ». Comme la fonction mère, `ztvovgn` prend en argument le nom d'utilisateur ainsi que le mot de passe en clair au format chaîne de caractères, et qu'ils sont passés concaténés et convertis en unicode à `ztcedchk`, on peut déduire que c'est cette fonction qui transforme les deux chaînes au format unicode (via `lxsNormStr`), les concatène et les retourne.
- Pour finir, le mot de passe est retourné au format ASCII. La conversion est certainement réalisée par la fonction `ztuc8tx`, au moyen de deux appels successifs à `ztuc4tx`.
- Le premier appel à `ztcedchk` n'est pas accompagné par un appel à `ztcedgks`. C'est-à-dire que la chaîne est chiffrée sans que la clef de chiffrement soit définie. Cette clef est donc certainement définie ailleurs et fixe.

Pour aller plus loin, il est alors nécessaire de passer à des méthodes classiques :

- utiliser `gdb` pour observer les zones mémoire traversées ;
- utiliser des outils tels qu'`objdump`, ou mieux, `IDA` pour désassembler le code ;
- et surtout écrire des bouts de programmes utilisant les fonctions étudiées qui seront *linkés* avec les bibliothèques Oracle de sorte à étudier leur comportement.

Et finalement comment fait-on pour calculer le hachage ?

Après tous ces efforts, voici enfin l'algorithme recherché :

- Le nom de l'utilisateur ainsi que le mot de passe sont concaténés dans une chaîne de caractères.

→ Celle-ci est convertie en unicode et toutes les lettres minuscules sont transformées en majuscules.

→ La chaîne résultante est chiffrée en utilisant DES en mode ncbc, avec un vecteur d'initialisation nul, et comme clef 0x123456789abcdef.

→ Le vecteur d'initialisation mis à jour par l'opération précédente est utilisé comme clef, pour rechiffrer la chaîne, de la même manière.

→ Le nouveau vecteur d'initialisation mis à jour contient le mot de passe.

Et c'est gagné ! Un petit programme de démonstration :

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <dlfcn.h>
#include <link.h>

/* DES en mode cbc, stocke l'iv dans la variable iv */
extern int ztcedchk(char * key_schedule, unsigned char * entree, int len, char *
sortie, int * iv);
/* initialise le key schedule en fonction de la clef. */
extern int ztcedgks(unsigned int * key, unsigned char * schedule, int val);

/* met en majuscules */
char upper(char b)
{
        if( (b>='a') && (b<='z') )
                b -= 0x20;
        return b;
}

/* conversion d'indianite sur des shorts */
#define POS(x) ( ((x)&0xFFFFFFFE) + (1-((x)&1)) )

void my_ztvovgn(char * output, char * password, int pwdlen, char * username, int
usernamelen)
{
        unsigned char schedule[1024]; /* key schedule */
        unsigned char sortie[500]; /* buffer contenant les flots chiffres, non
utilise */
        unsigned int iv[2]; /* les iv, declares en int pour plus de simplicite */
        unsigned int i;
        unsigned int len;

        unsigned char new_username[0x1F0]; /* pour stocker la chaine unicode */

        memset(new_username, 0, 100 );

        /* conversion unicode du pauvre ... */
        for(i=0;i<usernamelen;i++)
                ((unsigned short *)new_username)[ POS(i) ] = upper(username[i]) ;
        for(i=0;i<pwdlen;i++)
                ((unsigned short *)new_username)[ POS(i + usernamelen) ] =
upper(password[i]) ;

        printf("%s:", username);

        /* calcul de la longueur a casser en DES, un peu etrange, mais
probablement a cause des conversions d'indianite */
        len = (usernamelen + pwdlen);
        len = (((len-1)|3)>>1)+1;

        /* la premiere clef est initialisee */
        iv[0] = 0x01234567;
        iv[1] = 0x89abcdef;
        ztcedgks(iv, schedule, 1);

        /* premier chiffrement */
        ztcedchk(schedule, new_username, len, sortie, iv);

        /* la seconde clef est initialisee */
        ztcedgks(iv, schedule, 1);

        /* second chiffrement */
        ztcedchk(schedule, new_username, len, sortie, iv);

        printf("%x%x\n", iv[0], iv[1]);
}

int main(void)
{
        unsigned char stuff3[200];
        unsigned char * username = "system";
        unsigned char * password = "thales";

        my_ztvovgn(stuff3, password, strlen(password), username,
strlen(username));
        return 0;
}

$ gcc -Wall -o test -L/u01/app/oracle/OraHome_1/lib/ test.c -lclntsh -lsqlplus -
lnnz10 && LD_LIBRARY_PATH=/u01/app/oracle/OraHome_1/lib/ ./test
system:9eedfa0ad26c6d52
```

Références

[1] Documentation DTrace officielle : **http://docs.sun.com/app/docs/doc/817-6223**

[2] Adam Leventhal's *weblog* : **http://blogs.sun.com/ahl**

[3] *The Observation Deck* : **http://blogs.sun.com/bmc**

[4] $<*blog* : **http://blogs.sun.com/mws**

[5] Le *post* de Bob Baldwin, créateur de l'algorithme : **http://groups.google.com/group/comp.security.misc/msg/83ae557a977fb6ed?output=gplain**

[6] Oracle *Password Tools* : **http://www.red-database-security.com/whitepaper/oracle_password_cracker.html**

[7] Ressources pour John the Ripper : **http://www.banquise.net/misc/patch-john.html**

La simulabilité des tests statistiques

Les tests statistiques constituent un outil puissant et omniprésent. Utilisés lors de processus décisionnels, ils sont à la base de la plupart des activités humaines. Choisir, décider, commander sont des actes qui tous interviennent à la suite d'un ou plusieurs tests statistiques. La question alors se pose de la fiabilité de ces tests, de la confiance que l'on peut leur accorder. Winston Churchill ne déclarait-il pas qu'il existait trois types de mensonges : les simples mensonges, les mensonges sacrés et les statistiques. On peut également s'interroger sur la dépendance des décideurs vis-à-vis des tests statistiques. Dans le contexte plus spécifique de la sécurité, l'usage de ces tests n'est-il pas à double tranchant ? Autrement dit, un attaquant ne peut-il pas précisément utiliser contre le décideur, les tests statistiques que ce dernier a adopté pour étayer son processus de décision. Cette situation, si elle est avérée, serait particulièrement critique dans le domaine de la sécurité. L'objectif de cet article est d'expliquer ce qu'est un test statistique et de montrer qu'il est effectivement possible, pour un attaquant, de flouer l'utilisateur mettant en œuvre de tels tests, dès lors que ce dernier les a rendus publics : on parle alors de « simulabilité des tests ». Le cas du contrôle d'aléa, crucial en cryptologie, sera considéré à titre d'illustration.

Introduction

Qu'y a-t-il de commun entre :

- la détection de virus ;
- la recherche de contenu stéganographique ;
- la traque des délits d'initiés [7] ;
- l'analyse d'images ;
- le calcul d'audimat ;
- le contrôle de la qualité aléatoire d'une suite ;
- la cryptanalyse d'un système de chiffrement ;
- les prévisions météorologiques ;
- les analyses politiques de soirées électorales ;
- le lancement d'un produit de consommation ;
- la définition de politique de santé publique ;
- les tests d'efficacité ou d'innocuité d'un médicament ou d'un produit ;
- ... ?

En fait, derrière cet inventaire à la Prévert, le dénominateur commun est la prise d'une décision fondée sur les résultats d'un ou plusieurs tests statistiques. Ces outils mathématiques ont pris une importance telle qu'ils sont devenus incontournables. Peut-être trop, car ceux qui les utilisent pour fonder leurs décisions – le meilleur exemple est l'état de forte dépendance des décideurs vis-à-vis des sondages – en viennent à oublier que non seulement les statistiques ne sont pas une science exacte mais également qu'il est possible de les manipuler. La meilleure façon de résumer les choses est de citer Abe Burrow : « La raison des statistiques est de vous donner raison ». La pire des situations est celle où un manipulateur (dans un contexte SSI, l'attaquant ou le vendeur de produit de sécurité) utilise les statistiques pour donner une image « adéquate » de la réalité qu'il veut défendre.

Afin de bien faire comprendre les choses, considérons le problème suivant généralement posé à la fin d'un cours d'introduction aux statistiques[1] :

À la suite d'un sondage, sur un échantillon de N personnes d'une commune, p_{obs} % sont favorables à l'implantation d'une grande surface. Pour quelles valeurs de N et de p_{obs}, le résultat ne contredit-il pas l'hypothèse selon laquelle un habitant de la commune sur deux est favorable à l'implantation de cette grande surface ?

Un rapide et simple calcul permet alors de montrer que si le sondeur trouve un échantillon de n = 200 personnes tel que p_{obs} = 49 % alors l'hypothèse est vraisemblable (le maire peut alors décider d'autoriser l'implantation) en revanche si n = 400 et p_{obs} = 48 %, alors on rejette l'hypothèse (le maire refuse l'implantation). On voit donc que la procédure de test en statistique est très sensible à des paramètres comme la taille de l'échantillon, la mesure observée sur cet échantillon... Selon la façon, volontaire ou non, de préparer l'échantillon, les statistiques vous permettent de prouver une chose et son contraire. Nous laisserons le soin au lecteur d'imaginer alors tout ce qu'il est possible de faire.

Le problème précédent et surtout les résultats qui peuvent ensuite être donnés sont en fait incomplets. Il manque une donnée très importante : la valeur du risque éventuel attaché à ma décision, c'est-à-dire le risque que je cours de faire un mauvais choix. Or cette donnée est très importante car c'est elle qui permettra d'évaluer la qualité, la pertinence de ma décision. Dans le problème précédent, les deux résultats possibles ont été établis avec un risque de 5 % (cela veut dire que si je fais le test 100 fois, je prendrai une mauvaise décision 5 fois en moyenne). Si maintenant je veux réduire ce risque à 1 %, dans les deux cas de figures (n = 200 et n = 400), le test conduit à accepter l'implantation. Pour la rejeter, il faut un échantillon de n = 500 personnes et

[1] Sa résolution en détail est laissée au lecteur, à titre d'exercice -:).

Eric Filiol
Ecole Supérieure et d'Application des Transmissions
Laboratoire de virologie et de cryptologie
efiliol@esat.terre.defense.gouv.fr

p_{obs} = 44 %. S'agissant de résultats à destination du grand public, la donnée du risque – qui est un choix du testeur, en fonction de critères liés aux implications de la décision finale[4] – n'est que très rarement précisée. D'une manière générale, il faut consulter les documents techniques décrivant le protocole des tests.

Ce petit problème, même s'il est basique, illustre bien le problème des statistiques : la nécessité constante de devoir interpréter les résultats et conserver à l'esprit qu'une erreur est toujours possible.

Voyons maintenant plus précisément comment on construit un test statistique et comment cela fonctionne.

Qu'est ce qu'un test statistique ?

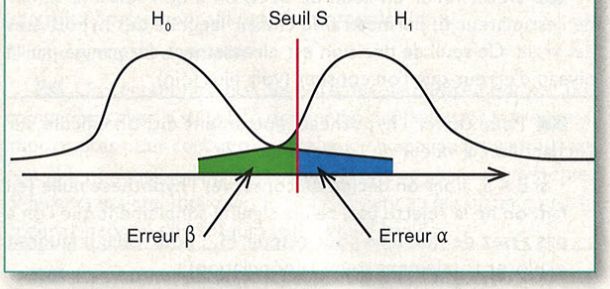
Il existe plusieurs types de tests statistiques : unilatéraux, bilatéraux, paramétriques ou non paramétriques, d'adéquation... [1,2], mais tous se ramènent à une forme générique de tests appelés « tests d'hypothèses ». Présentons la philosophie générale de procédure de décision. Précisons tout d'abord que les tests sont utilisés du fait de l'incapacité de l'observateur et du décideur à acquérir une connaissance complète d'une population.

On va donc utiliser des échantillons (provenant de sondages), c'est-à-dire, des sous-ensembles réduits de cette population. À partir de ces sous-ensembles, on va, en vertu de lois de comportement de ces échantillons[2] que l'on appelle de manière générique la « distribution d'échantillonnage », prendre une décision concernant la population que l'on souhaite étudier. Ce domaine de la statistique se dénomme « statistique inférentielle ». Il s'agit d'un raisonnement par induction.

Un test statistique est un protocole permettant de décider entre deux hypothèses : **l'hypothèse nulle** et **l'hypothèse alternative**. L'outil principal utilisé est **l'estimateur** E. Il s'agit d'une mesure faite sur les individus d'un échantillon. Selon l'une ou l'autre des hypothèses du test, cet estimateur est modélisé par une loi de probabilité différente. Un test d'hypothèses revient donc à décider, en vertu des valeurs d'un estimateur sur un ou plusieurs échantillons, quelle est la loi qui gouverne la population que l'on étudie. Considérons la figure 1 décrivant les deux hypothèses et leur loi de probabilité respective (les deux courbes) pour l'estimateur E considéré.

Un test se construit de la manière suivante[3]. Nous allons prendre comme exemple l'étude d'une suite binaire. Cela nous servira dans la partie suivante consacrée à la simulabilité des tests. Pour bien comprendre les choses, la population que l'on veut étudier consiste en toutes suites binaires qu'un système de chiffrement par flot M peut produire [10]. Il n'est bien sûr pas possible d'étudier cette population en totalité.

Figure 1 *Description graphique générale d'un test d'hypothèses*



On formule tout d'abord les hypothèses par rapport à un estimateur E donné (variable aléatoire d'un échantillon à un autre) censé être représentatif, dans son comportement moyen, d'un paramètre théorique d'une population. Le choix de l'hypothèse nulle doit être fait avec soin. En règle générale, elle correspond à l'hypothèse selon laquelle on est conduit à conserver la valeur présumée du paramètre étudié pour la population. L'hypothèse nulle est généralement celle que l'on ne rejette qu'à contrecœur.

Pour notre exemple, l'estimateur E sera le nombre de 0 dans une suite binaire de longueur N. L'hypothèse H_0 décrit le fait que les suites produites par la machine M contiennent en moyenne autant de 0 que de 1 (propriété importante en cryptologie). L'hypothèse H_1 affirme au contraire que la suite contient un biais en faveur des 1 (plus de 1 que de 0).

On choisit un seuil de signification, noté α, correspondant au risque de rejeter H_0 alors que cette hypothèse est vraie. Le choix de ce seuil est assez délicat[4], mais c'est également lui qui permet (éventuellement) de jouer sur le résultat du test. Nous verrons un peu plus loin le problème des erreurs attachées à un test. Choisissons ce seuil égal à 1%.

Il faut ensuite déterminer la loi de probabilité correspondant à la distribution d'échantillonnage pour chaque hypothèse. En

[2] La détermination et l'étude de ces lois relèvent du domaine de la théorie des probabilités.

[3] Nous décrivons ici le modèle générique auquel se ramène la quasi-totalité des tests statistiques.

[4] D'une manière générale, ce risque est déterminé par les conséquences d'une éventuelle erreur de décision. Ces conséquences ne sont pas les mêmes selon qu'il s'agit de l'évaluation des effets secondaires sur l'organisme d'une substance susceptible d'être commercialisée ou bien d'estimer quel choix les électeurs vont faire lors d'une consultation politique locale.

d'autres termes, quel est le modèle mathématique décrivant le comportement moyen de l'estimateur E choisi, d'un échantillon à l'autre, selon que c'est H_0 ou H_1 qui est valide. Précisons que si on est toujours en mesure de le faire pour l'hypothèse nulle, ce n'est pas le cas pour H_1 dont la loi est assez souvent inconnue. Dans notre exemple, la théorie dit qu'en moyenne le nombre de bits égaux à 0 d'une suite aléatoire (hypothèse nulle) de longueur N vaut N/2.

■ Déterminer un seuil de décision S qui, selon la valeur de l'estimateur E, permettra de choisir laquelle des hypothèses est vraie. Ce seuil de décision est directement déterminé par le niveau d'erreur que l'on consent (voir plus loin).

■ Pour tester l'hypothèse, autrement dit, on calcule sur l'échantillon la valeur de l'estimateur E et :

- si E < S, alors on décide de conserver l'hypothèse nulle (en fait, on ne la rejette pas, ce qui signifie simplement que l'on a pas assez de données pour retenir H_0 ; pour cela, il faudrait explorer totalement toute la population) ;
- sinon (E > S), alors on décide que c'est l'hypothèse alternative qui vraisemblablement s'applique.

Le lecteur aura remarqué l'usage de termes relativement vagues tels que : en moyenne, vraisemblablement... Cela traduit d'une part le fait que l'on est condamné à évaluer une population à partir d'une infime partie de cette dernière et d'autre part que d'un échantillon (partie) à un autre, les résultats observés peuvent varier quelquefois de manière importante[5]. C'est la raison pour laquelle on utilise la notion de distribution d'échantillonnage, qui modélise le comportement général des échantillons (donc leur variabilité pour un estimateur E donné). Ainsi, si on étudie 1000 suites de longueur N = 10000[6], la théorie montre qu'il est normal d'en trouver en moyenne 22 ayant plus de 5200 bits à 1 (cela correspond à 0.0228 %).

Deux risques peuvent alors entacher ma décision. Ils sont résumés dans le tableau ci-dessous.

Probabilités des deux types d'erreur

Décision	H_0 vraie	H_1 vraie
Accepter H_0	$1 - \alpha$	β
Accepter H_1	α	$1 - \beta$

Le risque α (risque dit « de première espèce ») représente la probabilité de rejeter l'hypothèse H_0 alors que cette dernière est vraie. Ce risque correspond sur la figure 1 à la zone en bleue. Si l'on reprend l'exemple précédent des 1000 suites de longueur N = 10000, le test a échoué 23 fois (le seuil a été fixé à S = 5200, α vaut donc 2.28 %) et l'on a rejeté H_0 alors que H_0 était pourtant vraie. Une erreur de décision a été commise lors de 23 tests sur 1000. La probabilité $1 - \alpha$ correspond à la surface de la courbe modélisant H_0 et située à gauche du seuil S : c'est la zone d'acceptation du test (accepter H_0 quand cette dernière est vraie) encore appelée « zone de convergence » du test.

Le risque β (risque dit « de seconde espèce ») représente la probabilité de conserver l'hypothèse H_0 alors qu'elle est fausse (zone verte sur la figure 1). Dans le contexte cryptologique de notre exemple, cela conduit à considérer que le système de chiffrement est bon alors qu'il est au contraire d'une qualité insuffisante. C'est donc dans notre cas, le risque le plus dangereux.

Dans beaucoup de situations (en cryptologie ou dans d'autres domaines), comme la loi de probabilité gouvernant H_1 est inconnue, il est impossible de calculer le risque β. Cependant, dans certaines circonstances très particulières, la loi de H_1 peut être connue d'un nombre limité de personnes, mais pas du testeur. C'est cette connaissance qui est à la base de la simulabilité des tests statistiques.

La simulabilité des tests : application au contrôle d'aléa

Il existe deux formes de simulabilité : l'une ne dépend pas des paramètres du ou des tests, c'est la **simulabilité forte**. L'autre, au contraire, dépend des paramètres du ou des tests. C'est la **simulabilité faible**. Ces résultats et les preuves mathématiques sont détaillés dans [4].

La simulabilité forte

Donnons tout d'abord une définition du concept.

■ Simulabilité forte d'un test statistique

Soit une propriété P et soit un test T destiné à vérifier si P est valide pour une population U donnée. Simuler fortement ce test, c'est concevoir la population U de manière à ce que T décide systématiquement, au risque d'erreur près, que P est vérifiée sur U, mais qu'il existe un test T' tel que ce dernier décide du contraire. De la même manière, on dira que l'on simule fortement t tests (T_1, T_2... T_t) si leur application conjointe conduit à décider que P est vraie sur U mais ne l'est plus si on considère un (t+1)-ème test T_{t+1}.

En termes de sécurité, la simulabilité forte des tests prend une importance capitale, notamment quand le testeur n'utilise que t tests pour évaluer une population U proposée par un tiers qui lui seul connaît le (t+1)-ème test T_{t+1}. Toujours dans le contexte de la sécurité, on peut alors résumer les principaux cas de la manière suivante :

[5] Pour comprendre cette variabilité naturelle, considérons l'exemple suivant : on souhaite connaître la taille moyenne de la population européenne. Il est facile alors de voir qu'un échantillon de norvégiens ne donnera pas les mêmes résultats que pour un échantillon de personnes originaires du sud de l'Europe.

[6] Pour clarifier les choses, chaque suite représente un échantillon. Les bits de la suite sont les individus. On cherche alors à tester que la population des bits produits par le système de chiffrement respecte le principe d'équilibre (production d'autant de bits à 0 que de bits à 1). Considérer 1000 suites (échantillons) revient à faire le test 1000 fois.

- le testeur et le tiers ne connaissent pas le test T_{t+1}. C'est le cas où un produit de sécurité est mis sur le marché et où tout le monde pense qu'il est bon (la propriété P est vérifiée sur U). Quand un chercheur imagine le test T_{t+1} alors le produit est déclaré vulnérable. En cryptanalyse, le travail consiste précisément à imaginer des tests mettant en lumière des faiblesses exploitables.

- le testeur ignore T_{t+1} mais pas le tiers. C'est le cas où le tiers a mis une trappe dans le système. Le test T_{t+1} constitue alors un secret qui permet au tiers seulement de mettre à mal la sécurité d'un produit. On parle alors de trappe dans le produit. Une instance de ce cas a été présentée lors de la conférence SSTIC 2003 [10] : la trappe est une structure algébrique. Indétectable avec les tests classiques, elle le devient en appliquant un test de plus, connu, le **test du rang**. En modifiant alors le système, dans le but de simuler également le test du rang, ce dernier conclut à l'absence de biais après modification du système qui est alors décidé bon pour le service.

Afin de montrer le risque gravissime de la simulabilité de tests, nous allons considérer le cas du contrôle d'aléa. En d'autres termes, la qualité cryptologique affichée d'un système. Le but est de prouver que, précisément, on ne peut avoir aucune garantie dans ce domaine.

En cryptologie, un bon système de chiffrement doit exhiber les meilleures propriétés aléatoires possibles. En effet, le but ultime est que celui qui intercepte un texte chiffré ne puisse pas le distinguer d'un aléa dit « vrai »[7]. Mais qu'est-ce que l'aléa ? Comment le définir ? Considérons une suite binaire. Une approche naïve consisterait à considérer un premier test où l'estimateur prend en compte le nombre de bits à 1 ou à 0. Une suite aléatoire contient donc autant de 0 que de 1. Avec ce seul test, la suite suivante est correcte :

```
0101010101010101010101010101010101.....
```

Il est facile de voir que la suite est tout sauf aléatoire. Les motifs 00 et 11 sont absents. On peut alors considérer le test consistant à vérifier que tous les motifs 00, 01, 10 et 11 sont présents.

La suite

```
0010011100100111001001110010011100100111001001110010011...
```

satisfait les deux tests mais elle reste inutilisable en cryptographie car certains motifs de longueur 8 sont absents. Le principe peut alors s'appliquer indéfiniment sur des motifs de longueur quelconque et pour chaque biais mis en évidence : chaque fois qu'une faiblesse est détectée, un nouveau test est conçu pour la détecter. En retour, le système peut être modifié de sorte à introduire une nouvelle faiblesse (biais), exploitable par celui qui la connaît, mais que les tests connus ne signaleront pas.

Alors, comment définir ce qu'est une suite aléatoire ? En fait, l'aléa n'est pas une notion absolue, mais relative à une batterie de tests. Ce qui est aléatoire une fois que l'on a appliqué t tests ne le sera peut-être plus pour un (t+1)-ème. Et il est toujours possible de trouver un test « de plus » [4]. Ce sera une certitude si celui qui introduit une trappe dans un système (voir [10]) en prenant soin d'être compatible avec tous les tests communément utilisés pour déclarer une suite comme cryptographique bonne. L'outil le plus connu est la suite STS [8] distribuée par le Département du Commerce (USA). Elle sert de référence pour mesurer la qualité cryptographique d'une suite. Mais, les tests mis en œuvre dans cette suite peuvent être simulés fortement de sorte à produire une suite bonne pour un usage cryptographique.

Citons quelques exemples :

- Un certain nombre de systèmes de chiffrement par flot compatibles avec la suite STS ont exhibé des faiblesses statistiques, importantes pour certains d'entre eux, en appliquant un nouveau test [5], non inclus dans cette suite. Comme plusieurs systèmes différents étaient impliqués, le résultat tient au fait qu'un nouveau critère, inconnu des auteurs, est à considérer.

- Le système « trappé » proposé par E. Wegrzynowski [10] passe avec succès les tests de la suite STS.

- À titre de challenge, une suite de 2^{26} bits a été générée [6] de telle sorte à contenir une faiblesse permettant une cryptanalyse opérationnelle. Cette suite a pourtant passé les tests STS. Le challenge consiste à imaginer un test qui permettra de prouver que la suite n'est pas cryptographiquement bonne.

Le lecteur remarquera que ce type de simulabilité ne considère à aucun moment les paramètres caractéristiques d'un test (risques d'erreur notamment). D'où l'appellation de simulabilité forte. Mais quand un tiers, qui souhaite flouer le testeur, ne dispose pas d'un test inconnu de ce dernier (cas de la simulabilité forte), il est intéressant de considérer un autre type de simulabilité qui consiste à jouer directement sur les paramètres des tests : c'est la simulabilité faible.

La simulabilité faible

Précisons tout d'abord le concept.

Simulabilité faible d'un test statistique

Soit une propriété P et soit un test T destiné à vérifier si P est valide pour une population U donnée. Simuler faiblement ce test, c'est introduire dans la population U une nouvelle propriété P' modifiant partiellement la propriété P, de manière à ce que T décide systématiquement, au risque d'erreur près, que P est vérifiée sur U.

Par rapport à la simulabilité forte, là on travaille avec les mêmes tests que ceux utilisés par le testeur. On va donc introduire un biais mais que la sensibilité des tests utilisés ne permettra pas de détecter, en vertu des risques généralement utilisés par le testeur.

[7] Par aléa vrai, on entend un aléa totalement non reproductible généralement produit par des phénomènes naturels (oscillations d'une résistance thermique, émissions d'une particule radio-active...). Mais là également, on ne sait finalement pas le caractériser autrement que d'une manière relative, comme expliquée dans cet article. Le lecteur intéressé par la conception de la notion d'aléa en physique théorique (controverse Einstein-Bohr sur la théorie des quanta : « Dieu joue-t-il ou non aux dés ? ») pourra consulter [9].

La propriété P' de la définition s'oppose en règle générale à la propriété P. Elle constitue précisément une vulnérabilité que le tiers veut exploiter sans que le testeur s'en rende compte lors de l'évaluation. Mettre en œuvre la simulabilité faible est assez délicat et réclame de bien connaître la structure et les propriétés mathématiques du ou des tests à simuler, en particulier quand il y a plusieurs tests à simuler simultanément. La philosophie générale de la simulabilité faible consiste à introduire la propriété P' de sorte à ce que le ou les estimateurs E utilisés restent dans la zone de convergence. Lors de la phase de décision, rappelons que l'on regarde si E < S. Simuler faiblement le test, consiste alors grosso modo à jouer avec la valeur S - E tout en s'assurant qu'elle reste positive. Pour ce faire, on exploite les propriétés de la distribution d'échantillonnage tout en exploitant la marge offerte par les paramètres du test.

Pour illustrer, de manière, certes triviale, la simulabilité faible du test d'équilibre d'une suite (autant de 0 que de 1 dans la suite), considérons la suite suivante :

`10100100110110101110001000100110111O.....`

Cette suite est équilibrée. Le test d'équilibre conclura que la suite est bonne pour un usage cryptographique (si l'on ne considère que ce seul critère). Transformons alors la suite de la manière suivante (les modifications sont signalées entre parenthèses) :

`1(0)01(1)001(0)0011(1)01101(0)011100(1)0100010(0)01101110(1).....`

Le biais introduit (la propriété P') consiste à diviser la suite en sous-suites de taille croissante : 1, 2, 3... et ensuite à insérer alternativement 0 et 1 entre deux sous-suites. Un biais exploitable par un cryptanalyste a été inséré mais le (seul) test d'équilibre conduit à la bonne qualité de la suite.

Lorsque plusieurs tests sont à simuler faiblement et simultanément, le principe reste le même mais peut être considérablement plus technique et mathématique (voir [4] dans le cas de la détection virale). Enfin, notons que la simulabilité faible s'applique dans tous les domaines mettant en œuvre des tests statistiques. Le lecteur trouvera dans [6] une suite binaire simulée faiblement mais jugée comme cryptographiquement bonne par la suite STS [8].

Conclusion

La simulabilité des tests pose un problème crucial en sécurité : comment concevoir un produit de sécurité, sachant qu'il est impossible d'avoir des certitudes sur son efficacité réelle ? La population des événements décrivant en totalité ledit produit est d'une taille telle qu'une vision partielle et approchée, par des tests, est le seul moyen d'inférer ce que peut être la sécurité du produit au final. Dans le cas de la cryptologie, cela signifie qu'un cryptographe développe son produit en l'« état actuel des connaissances ». Il ne peut pas garantir, sinon à un risque d'erreur non nul près, que son produit offre un niveau de sécurité donné. Il faut cependant relativiser le propos. Les cryptographes effectuent des milliers de tests, et qui plus est des centaines de tests sur les résultats des tests. Le but, si le risque d'erreur ne peut jamais être nul, est de le faire tendre vers 0. D'autre part, l'art de l'ingénieur est peut être encore plus important que l'art du statisticien : l'expérience et le savoir-faire permettent d'envisager ce que la statistique ne peut qu'imprécisément décrire.

Le cas d'un produit de sécurité proposé par un tiers pose un tout autre type de problème. Le testeur ne peut évaluer le produit qu'avec les tests qu'il connaît. Le tiers qui propose le produit le sait et peut introduire une faiblesse volontaire (une trappe) indétectable par simulation des tests d'évaluation utilisés par le testeur. Connaître le mode de fonctionnement du produit (l'algorithme ou autre) n'est pas une garantie en soi, si les états internes de ce dernier décrivent une population impossible à appréhender sinon que par infimes parties. Le seul moyen est de protéger une partie des tests utilisés : aucun tiers ne peut alors les simuler.

Les deux ensembles de données proposés aux lecteurs [6] comme challenge permettront à ces derniers d'une part de pratiquer les tests statistiques et d'autre part, à l'issue, de mieux comprendre les problèmes posés par leur simulabilité.

Références

[1] DODGE, Y., *Premiers pas en statistique*, Springer Verlag, 2nde édition, 2005.

[2] DODGE, Y., *Statistique : dictionnaire encyclopédique*, Springer Verlag, 2nde édition, 2004.

[3] FILIOL, E., « Le chiffrement par flot », *Journal de la sécurité informatique* MISC 16, novembre 2004.

[4] FILIOL, E., *Statistical Modelling of Viral Detection Undecidability*, à paraître, 2006.

[5] FILIOL, E. « *A New Statistical Testing for Symmetric Ciphers and Hash Functions. Proceedings of the 4th International Conference on Information and Communication Security 2002* », Lecture Notes in *Computer Science*, Springer, 2002.

[6] **http://www-rocq.inria.fr/codes/Eric.Filiol/Misc/misc22_s_fort** (taille 64 Mo)
et **http://www-rocq.inria.fr/codes/Eric.Filiol/Misc/misc22_s_faible** (taille 64 Mo).

[7] PONTIER, M. et GRORUD, A., « Comment détecter les délits d'initiés », in Comptes-rendus de l'Académie des Sciences, série I, tome 324, pp. 1137-1142, décembre 1997.

[8] *A Statistical Test Suite for the Validation of Ramdom Number Generator and Pseudo-random Number Generator for Cryptographic Applications*, Revised NIST Special Publication 88-22, National Institute of Standard and Technology, US Commerce Department's Technology Administration, 2000. Le document ainsi que le logiciel de test (version 1.8) sont disponibles sur **http://csrc.nist.gov/rng/rng2.html**.

[9] SCHIPP, P.A. (éd.), *Albert Einstein : Philosopher Scientist*, Tudor Publishing, 1970.

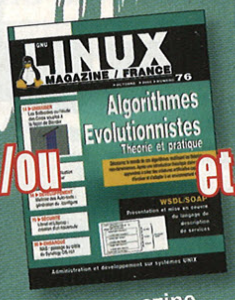
[10] WEGRZYNOWSKI, E., « Des trappes dans les clefs », in Actes de la conférence SSTIC 2003, pp. 248-260, juin 2003.

Offres de couplage !

Lisez-vous régulièrement :

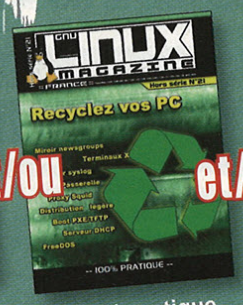
Le magazine 100% sécurité informatique

et/ou



Le magazine 100% Linux

et/ou

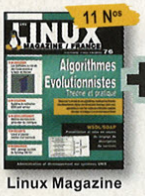


100% pratique

et/ou

Apprivoisez votre pingouin !

Si oui, ces offres d'abonnement à tarif préférentiel vous sont destinées.



Linux Magazine

+

Linux Magazine hors série

~~108,80~~

79 €

Economie : 29, 80 €

Linux Magazine

+

Misc

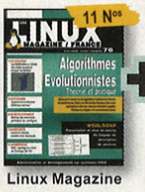
+

Linux Magazine hors série

~~153,50~~

105 €

Economie : 48, 50 €



Linux Magazine

+

Misc

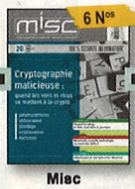
~~115,40~~

83 €

Economie : 32, 10 €

Linux Magazine

+



Misc

+

Linux Magazine hors série

+

Linux Pratique

~~189,20~~

129 €

Economie : 60, 20 €

Bon de commande à remplir et à retourner à :

Diamond Editions - Service des Abonnements/Commandes 6, rue de la Scheer B.P. 121 - 67603 Sélestat Cedex

OUI, je m'abonne et désire profiter des offres spéciales de couplage			
Je coche la référence de l'offre :	Prix	Qté.	Total
☐ 11 Nos Linux Mag. + 6 Nos Linux Mag HS	79 €		
☐ 11 Nos Linux Mag. + 6 Nos MISC	83 €		
☐ 11 Nos Linux Mag. + 6 Nos MISC + 6 Nos Linux Mag HS	105 €		
☐ 11 Nos Linux Mag. + 6 Nos MISC + 6 Nos Linux Mag HS + 6 Nos Linux Pratique	129 €		
OFFRES VALABLES UNIQUEMENT EN FRANCE MÉTRO*	TOTAL		

*Pour les tarifs étrangers, consultez notre site : www.ed-diamond.com

4 façons de vous abonner :

- ☒ par courrier postal en nous renvoyant le bon ci-dessous
- ☒ par le Web, sur **www.ed-diamond.com**
- ☒ par téléphone, entre 9h-12h et 14h-17h au 03 88 58 02 08
- ☒ par fax au 03 88 58 02 09 (CB)

1	Voici mes coordonnées postales
	Nom :
	Prénom :
	Adresse :
	Code Postal :
	Ville :

2	Je joins mon réglement :
☐	Je règle par chèque bancaire ou postal à l'ordre de Diamond Editions*
☐	Paiement par carte bancaire :
☒	N° Carte :
☒	Expire le : __/__/
☒	Date et signature obligatoire : __/__/ 200_

→ www.ed-diamond.com

Retrouvez et commandez sur notre site les précédents numéros de Misc (1 à 21).

Notre moteur de recherche vous permet de retrouver parmi nos parutions les articles susceptibles de vous intéresser !

DIAMOND editions

MISC
est édité par Diamond Editions
B.P. 121 - 67603 Sélestat Cedex
Tél. : 03 88 58 02 08
Fax : 03 88 58 02 09
E-mail : lecteurs@miscmag.com
Abonnement : abo@miscmag.com
Site : www.miscmag.com

Directeur de publication : Arnaud Metzler
Rédacteur en chef : Frédéric Raynal
Rédacteur en chef adjoint : Denis Bodor
Conception graphique & mise en page : Katia PAQUET - Franck TOUSSAINT
Impression : Presses de Bretagne
Secrétaire de rédaction : Dominique Grosse
Responsable publicité : Véronique Wilhelm
Tél. : 03 88 58 02 08

Distribution :
(uniquement pour les dépositaires de presse)
MLP Réassort :
Plate-forme de Saint-Barthélemy-d'Anjou.
Tél. : 02 41 27 53 12
Plate-forme de Saint-Quentin-Fallavier.
Tél. : 04 74 82 63 04

Service des ventes : Distri-médias :
Tél. : 05 61 72 76 24
Service abonnement :
Tél. : 03 88 58 02 08

Dépôt légal : 2e Trimestre 2001
N° ISSN : 1631-9036
Commission Paritaire : 02 09 K 81 190
Périodicité : Bimestrielle
Prix de vente : 7,45 euros

Imprimé en France

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

MISC est un magazine consacré à la sécurité informatique sous tous ses aspects (comme le sytème, 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.

PEARL

Le spécialiste du périphérique informatique

Adaptateur IDE USB 2.0 externe

Grâce à ce kit composé d'un adaptateur IDE / USB et d'une alimentation, vous pourrez brancher votre disque dur ou n'importe quel périphérique IDE sur un port USB à chaud et ce, sans boîtier externe ! Vous pouvez ainsi disposer d'un moyen de sauvegarde très pratique et peu encombrant. Livré avec **un adaptateur pour connecter un disque dur 2,5"**
Réf.PE8193

Connectez vos disques durs en externe sans utiliser de boîtier!

39,90 € TTC

Réchauffe tasse USB avec tasse en acier

L'objet indispensable pour votre bureau. Votre café ou votre thé resteront bien plus **longtemps chaud**. Branchez le simplement sur un port USB de votre ordinateur et n'ayez plus peur de retrouver un café froid sur votre bureau. ▶Réchauffe tasse avec plaque chauffante (diamètre: 75mm), Led et interrupteur ▶Tasse thermo en acier (200 ml) avec couvercle ▶Longueur du câble USB : 150 cm **Réf.PE6641**

4,90 € TTC

Coffre de rangements pour 390 CD

Idéal pour ranger tous vos CD dans un espace minimum, ce coffre de rangement très solide offre une protection sure à vos données. Les pochettes dans lesquelles se glissent les CD reposent sur des rails et permettent ainsi de retrouver rapidement le CD recherché. ▶Pochettes de rangement de couleur blanche▶Dimensions : 490x200x240mm (lxhxp)
Réf.PE8666

29,90 € TTC

Support à roulettes

Ce support à roulettes **très stable** peut accueillir la quasi-totalité des tours existantes sur le marché. Il **se règle très facilement** (largeur entre 1 et 25,5cm). **Réf.PE7156**


Mise en situation

14,90 € TTC

Station multimédia USB

Voilà un petit boîtier pratique qui vous permet de regarder vos films, d'écouter vos musiques et de voir vos photos sans que votre disque dur externe ne quitte son boîtier. Branchez-le simplement sur la prise péritel de votre téléviseur pour profiter de vos fichiers. La prise USB à l'avant du boîtier accepte les **disques durs externe** mais aussi une **clé USB** ou un **lecteur de cartes**. Son format réduit vous permet de le transporter partout et donc de profiter à tout moment de vos données. ▶Se branche sur une prise péritel ou sur des prises Cinch ▶Vitesse de transfert : max. 1,5 Mo/s ▶Formats supportés: MPEG 1, 2 et 4 (compatible DivX), DAT, AVI, MP3, WMA, JPEG ▶Alimentation : adaptateur secteur ▶Supporte le son Dolby AC3 ▶Dimensions : 123x90x23mm ▶Inclus : télécommande, base, adaptateur secteur, câble AV (Cinch) et adaptateur péritel **Réf.PE3451**

99,90 € TTC

Webcam UBS 2.0 "Observer Cam"

Une webcam **haute définition** pour vos discussions entre amis. Votre image en **haute résolution** en mode vidéo et des images parfaites pour Internet, voilà ce que propose cette petite merveille. ▶Capteur: CMOS **SXGA** ▶Taux de rafraîchissement de l'image: 30 fps (VGA ou SXGA) ▶Résolution maximale: 1280x960 pixels (par interpolation) ▶Entièrement amovible (360°, 90° en vertical) ▶Câble USB Combo: USB et audio (150 cm) ▶Compression des images automatique ▶Balance automatique des blancs ▶Balance automatique des couleurs ▶Système requis: Windows 2000/XP et carte graphique compatible DirectX9.0 ▶Dimensions: 3,5x2,9x31mm (sans clip) **Réf.PE7548**

29,90 € TTC

Casio QV-R61

Cet appareil photo dispose d'un **zoom optique 3x** et d'un écran TFT de contrôle de 2". ▶**Capteur CCD de 6 millions de pixels** ▶Mémoire de 9Mo, possibilité d'insérer des cartes SD ou MMC ▶Résolutions supportées : jusqu'à 2816x2112 ▶Résolution vidéo : 320x240 (15 images/sec) ▶Focale : 8 à 24mm ▶Mise au point : 60cm à l'infini, mode macro : 10-70 cm ▶Retardateur : 10 sec, 2 sec ▶Flash : manuel, auto, anti yeux rouges ▶Alimentation : 2 accus AA NiMH 2100 mAh inclus) ▶Dimensions : 88x60x33 mm ▶Poids : 168g (sans batterie) ▶Compatible **Pictbridge** et **USB Direct Print Réf.PE187**

6 Megapixels

229,90 € TTC

Mini souris sans fil et sans pile

Grâce à ce petit bijou de technologie vous ne serez plus à court de jus. Profitez du confort d'utilisation d'une mini souris sans devoir changer de pile. ▶Mini souris à fréquence ▶Alimentation par induction ▶Pas d'interférences ▶Ne nécessite aucun pilote sous Windows Millenium/2000 et XP ▶Le principe d'induction ne fonctionne pas sur une surface métallique ▶Câble USB inclus
Réf.PE5818

24,90 € TTC

www.pearl.fr

Demandez gratuitement votre Catalogue 164 pages

PEARL Diffusion 6, rue de la Scheer
Z.I. Nord - B.P. 121 - 67603 SELESTAT Cedex

0,12 €/mn
N° Indigo 0 820 822 823

POUR L'EXTENSION.fr DE VOTRE SITE INTERNET

Vous avez dû vous contenter de :

paslechoix.fr au lieu de **monprojet.fr**

Très bientôt, l'Europe aura enfin son extension .eu

monprojet.eu

TENEZ-VOUS PRÊT !

Prenez une longueur d'avance avec OVH

L'extension ".eu" va permettre à l'ensemble des acteurs du web en Europe (professionnels et particuliers) de référencer leur site via cette nouvelle extension.
Offrez-vous la possiblité de choisir le nom de domaine ".eu" de votre site.

http://www.OVH.com/eu

OVH.COM

Votre nom de domaine
8.9€ HT/an
TOUT COMPRIS

Votre hébergement mutualisé
12€ HT/an
TOUT COMPRIS

Votre serveur dédié
Administration 100% via le web
69€ HT/mois
PLESK 7.5 RELOADED inclus
TOUT COMPRIS

SANS ENGAGEMENT DE DURÉE

RCS Roubaix - Tourcoing 424 761 419 00011 - 140 Quai de Sartel 59100 Roubaix - Théorem