XMeta : une approche bayésienne pour le computer forensics

Thomas DUVAL (DGA/CELAR, Supélec) thomas.duval@supelec.fr
Bernard JOUGA (Supélec) bernard.jouga@supelec.fr
Laurent ROGER (DGA/CELAR) laurent.roger@dga.defense.gouv.fr
Supélec: Avenue de la boulaie, BP 81127 F-35511 CESSON SEVIGNE FRANCE
DGA / CELAR: route de Laillé, F-35170 BRUZ FRANCE

Résumé

L'analyse forensique (ou analyse post-mortem) aide les administrateurs système et les enquêteurs en leur fournissant des outils et des techniques qui leur permettent de comprendre une attaque informatique, de restaurer un système sain et de trouver le(s) attaquant(s). Cette activité est aujourd'hui en grande partie empirique et repose sur l'expérience des enquêteurs.

Nous proposons dans cet article d'utiliser les réseaux bayésiens pour automatiser (sous certaines conditions) les investigations informatiques. Notre modèle, XMeta, est basé sur la connaissance de l'architecture et de la configuration des systèmes d'information, sur les indices accumulés tout au long de l'enquête et sur les résultats des enquêtes précédentes. Ces informations sont enregistrées dans une base de données mise à jour à chaque nouvelle enquête, et qui permet de proposer des hypothèses pour l'enquête en cours.

L'intérêt de notre méthode est ensuite illustré par l'étude d'une affaire bien connue : l'attaque du réseau de Tsutomu Shimomura par Kevin Mitnick.

mot-clés *: Analyse forensique, preuve informatique, délit informatique, réseaux bayésiens.*

Introduction

"Les attaquants sont-ils toujours en action ?" , "Où sont-ils ?", "Comment sont-ils entrés ?", etc. Même si les systèmes d'information sont de plus en plus sécurisés, la criminalité informatique augmente [CLU], et la demande en analyse forensique augmente d'autant. On peut définir l'analyse forensique comme étant "le processus de récupération et d'analyse de données compromettantes sur des systèmes (électroniques) d'information" [MP01]. L'analyse forensique permet notamment aux administrateurs système et aux enquêteurs de répondre aux trois questions posées au début de ce paragraphe.

Quand un enquêteur arrive sur les lieux d'un délit informatique, son premier travail consiste à collecter le maximum d'information sur l'organisation attaquée et sur les systèmes compromis. L'objectif est de connaître l'activité de l'organisation, l'architecture de son système d'information, les dommages observés, les personnes impliquées (administrateurs, utilisateurs, etc.). Les enquêteurs doivent aussi collecter et préserver les informations contenues dans les systèmes informatiques. En ce qui concerne ces dernières, les tâches les plus importantes sont de sauvegarder les données "volatiles" des systèmes compromis (comptes actifs, processus en cours d'utilisation, etc.) et de faire des images "bit-à-bit" des média de stockage, pour que les enquêteurs puissent les analyser par la suite dans leur laboratoire.

Ces analyses permettent de récupérer des indices sur l'attaque :

- présence de données ou logiciels spécifiques (comme un cheval de Troie),
- historique des commandes,
- dégradation de fichiers de configuration (comme l'ajout de lignes dans le fichier /etc/inetd.conf),
- utilisation de canaux réseaux cachés,
- fichiers cachés ou supprimés,
- etc.

Comme les systèmes informatiques sont généralement connectés en réseaux, les enquêteurs doivent rebondir de système en système en remontant la piste de l'attaque. Mises à part ces investigations techniques, les enquêteurs doivent aussi aborder le problème dans le contexte de l'accident : "Quelle est

l'activité de l'entreprise ciblée ?", "Qui a accès aux données ?", etc., ce qui peut permettre d'obtenir de nouveaux indices. Ces informations peuvent ensuite être confirmées par des témoignages et des interrogatoires.

Enfin, les enquêteurs doivent rédiger un rapport contenant toutes les observations et les preuves. Ce rapport doit être compréhensible par des non-informaticiens (par exemple par un tribunal) mais il doit également contenir toutes les informations techniques permettant de comprendre l'attaque et au besoin de pouvoir rejouer l'attaque et l'enquête.

Ce rapport doit inclure les éléments suivants :

- tous les média mis en cause dans l'attaque,
- la configuration de tous les systèmes impliqués,
- la chronologie de l'attaque et de l'enquête,
- les preuves (sous format électronique ou physique),
- l'identité des enquêteurs,
- la description des analyses effectuées,
- la description du scénario de l'attaque,
- etc.

Nous proposons un système permettant d'aider les enquêteurs dans leurs investigations en leur donnant des indices sur le(s) délit(s). Ce système est basé sur un système expert qui utilise les réseaux bayésiens comme moteur d'inférence.

Nous avons choisi l'approche bayésienne pour 3 raisons :

- elle peut modéliser des situations complexes,
- c'est une logique non binaire (probabiliste),
- le modèle est résistant quand les données sont incertaines ou incomplètes.

Nous examinerons dans la première section l'utilisation des réseaux bayésiens dans le domaine de la sécurité. Puis nous présenterons un bref état de l'art de l'utilisation des réseaux bayésiens en analyse forensique. La troisième partie propose une description de notre méthode, puis nous montrerons dans une quatrième partie comment utiliser notre système sur un cas concret. Enfin, nous conclurons dans la dernière section.

1 Les réseaux bayésiens

Les réseaux bayésiens (RB) sont très largement utilisés dans de nombreux systèmes expert, pour la classification de données, comme par exemple dans SpamAssassin [Teab], ou pour l'inférence comme dans certaines applications forensiques [BWC02, Chr02]. Les RB sont des graphes acycliques orientés où les nœuds sont des variables et les liens des relations de causalité pondérées par des tables de probabilités conditionnelles [Pop00],[NWL+04]. Un RB permet de modéliser des situations dont la perception est incomplète. Par exemple, un administrateur système constate qu'un serveur Web Apache (A) a été défiguré (D). Il y a deux causes possibles à cette situation (FIG. 1) :

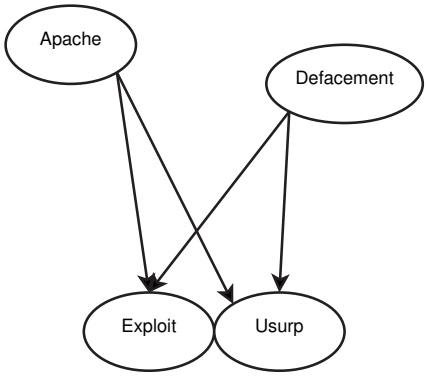

FIG. 1 – Un exemple simple de réseau bayésien

- l'attaquant a utilisé un exploit (E),
- l'attaquant a volé le mot de passe administrateur (U) (usurpation d'identité).

Un réseau bayésien bien formé va proposer des hypothèses sous forme de probabilités : il y a $X\%$ de chance que l'attaquant ait utilisé l'exploit (E) sur le serveur (A), la probabilité étant donnée par le théorème de Bayes :

$$P(E \mid A) = \frac{P(E, A)}{P(E)}$$

avec :

- $P(E \mid A)$ la probabilité que l'attaquant ait utilisé l'exploit (E) sachant que l'on a un serveur Web (A),
- $P(E, A)$ la probabilité d'avoir un exploit (E) et un serveur Web (A),
- $P(E)$ la probabilité de l'exploit (E).

La construction d'un réseau bayésien se fait en 3 étapes :

a) construction d'un graphe causal,

b) construction des tables de probabilités associées aux nœuds,

c) propagation de ces probabilités dans le réseau.

Deux méthodes peuvent être utilisées pour créer un réseau : la première consiste à interroger des experts du domaine qui détermineront les liens de causalité et qui proposeront des valeurs de probabilités. Cette façon de procéder peut être utilisée pour des réseaux de faibles dimensions. Pour un réseau de 10 nœuds, chacun de ces nœuds peut avoir jusqu'à 9 parents et donc avoir une table de probabilités associée de $9x9$, soit 81 valeurs possibles. Dans notre application, un réseau peut potentiellement avoir plusieurs centaines de nœuds (au maximum 7000 comme nous le verrons plus loin). Nous utilisons donc une autre méthode. A partir des résultats des enquêtes précédentes, l'algorithme d'apprentissage *k2* [CH92, NWL$^+$04] permet d'établir des relations de causalité entre nos variables et de calculer les valeurs de nos tables de probabilité. Ceci nous permet d'avoir un système qui s'actualise automatiquement au fûr et à mesure des cas traités.

Une illustration est donnée par le tableau 1. Dans notre exemple, ayant observé un défigurement $D = vrai$ et l'existence du serveur Web $A = vrai$, on en déduit que la probabilité que l'attaquant ait utilisé un exploit est

$$P(E = vrai \mid A = vrai, D = vrai) = 70\%$$

alors que la probabilité de l'usurpation est

$$P(U = vrai \mid A = vrai, D = vrai) = 35\%$$

On en conclut que l'attaquant a sans doute utilisé un exploit pour défigurer le serveur Web Apache.

L'inférence peut être descendante : des dégats observés (DoS, Defacement, etc.) vers les attaques (Exploit, Usurp) comme dans le cas de la figure 2. Elle peut être aussi montante : des attaques vers les logiciels pour savoir quel est le logiciel qui a la plus grande "chance" de s'être fait attaqué. Quand un utilisateur change la valeur d'un nœud, le changement des probabilités se propage à tout le réseau.

2 Analyse forensique et réseaux bayésiens

Plusieurs travaux ont été réalisés dans ce domaine. Dans [CJLM03], les auteurs analysent les systèmes pour récupérer toutes les informations utiles, dans le but de réduire le nombre d'information pertinentes et pour pouvoir découvrir, grâce à un réseau bayésien, toutes les communications réseau entre les différents systèmes

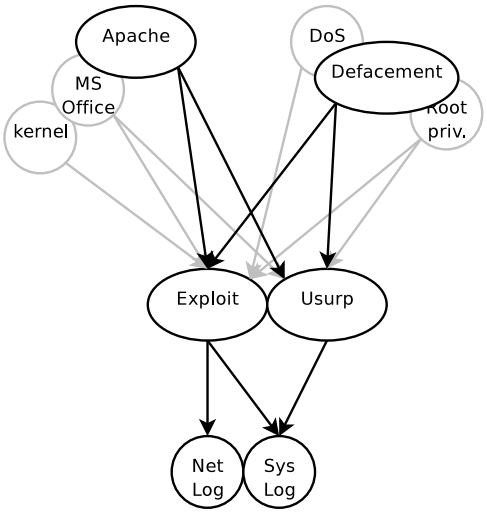

FIG. 2 – Un exemple de réseau bayésien plus complet

incriminés. Les résultats de ce projet ne concernent que les communications réseau. Nous proposons non seulement d'analyser les communications réseaux, mais aussi les autres données techniques (fichiers d'audit système) et non techniques (évènements du monde "réel", par exemple accès à un bâtiment).

Dans [LL00], les auteurs analysent avec des réseaux bayésiens des crimes non informatiques. Leur modélisation de la scène du crime met 4 éléments en évidence : les acteurs, les entités, les évènements et le contexte, ce qui leur permet d'émettre des hypothèses sur ces différents éléments. Ces hypothèses permettent d'analyser l'impact des indices sur les conclusions de l'enquête. Cela permet aux auteurs de quantifier la culpabilité des personnes incriminées et de préciser la manière dont a été perpétré le crime. La proposition est illustrée par son application à l'affaire Omar Raddad, qui a défrayé la chronique en 1994.

La finalité de nos travaux est la même : démasquer le ou les coupables. Mais notre proposition se concentre sur le guidage des enquêteurs dans leur choix des méthodes à utiliser pour trouver indices et preuves.

Les objectifs de notre système sont d'aider les enquêteurs à analyser les informations offertes par les systèmes d'exploitation, les logiciels et tout autre donnée utile pour inférer sur :

– les actions de l'attaquant et leur chronologie,
– comment et où les enquêteurs peuvent trouver des

$P(E \mid A, D)$	$D = vrai$		$D = faux$	
	$A = vrai$	$A = faux$	$A = vrai$	$A = faux$
$E = vrai$	70%	20%	50%	10%
$E = faux$	30%	80%	50%	90%
$P(U \mid A, D)$	$D = vrai$		$D = faux$	
	$A = vrai$	$A = faux$	$A = vrai$	$A = faux$
$U = vrai$	35%	10%	60%	50%
$U = faux$	65%	90%	40%	50%

TAB. 1 – Table des probabilités conditionnelles pour la variable Exploit E et Usurpation U

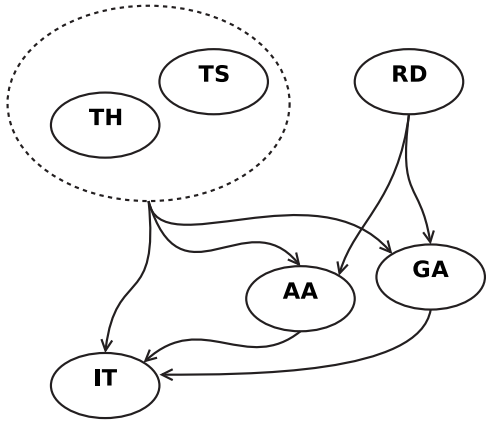

FIG. 3 – Structure d'un plan investigation

indices sur un système particulier,
– le profil du ou des attaquants,

3 Notre proposition

3.1 Le système XMeta

XMeta permet de travailler sur des "Plans d'Investigation" (IP, *Investigation Plan*). Un *IP* est un réseau bayésien qui représente un système compromis par une attaque particulière. Il est construit, au début de l'analyse d'un système, à partir de la configuration de celui-ci. Sa structure est présentée en figure 3. Les variables qui le composent (40 à 50 nœuds en moyenne) sont choisies parmi les six types suivants :

– le système cible (*Targeted Hardware*) : **TH**,
– les logiciels cibles (*Targeted Softwares*) : **TS**,
– les dégâts observés (*Reported Damages*) : **RD**
– les attaques génériques (*Generic Attacks*) : **GA** (il s'agit des 21 types d'attaques mis en évidence par la DGA, voir annexe),
– les actions complémentaires (*Additional Actions*) : **AA** (12 types d'actions, répertoriées par la DGA, pouvant accompagner une attaque, voir annexe),
– la source de l'attaque (*Source Address*) : **SA** (il s'agit d'un indicateur permettant de préciser si la source est locale (le même système), interne (le même réseau) ou externe),
– les techniques d'investigation à utiliser (*Investigation Techniques*) : **IT**.

L'enquêteur commence par saisir la configuration du système (**TH** et **TS**) et les dégâts constatés (**RD**). Les valeurs qui lui sont proposées sont extraites de la base de vulnérabilité ICAT qui recense aujourd'hui plus de 7000 logiciels [teaa]. L'*IP* est construit à la demande en fonction des valeurs choisies, les liens de causalité et les probabilités étant estimées à partir du résultat des enquêtes précédentes (utilisation de l'algorithme d'apprentissage *k2* [CH92, NWL$^+$04]). Le réseau peut ensuite proposer des hypothèses (utilisation de l'algorithme d'inférence approximative *Likelihood Weighting* [FC89]) : attaques **GA** ayant permis de pénétrer dans le système, éventuelles actions complémentaires **AA**, non obligatoires mais qui peuvent aider l'attaquant à atteindre son objectif (comme l'envoi d'un message "I 0wn y0u !" par mail ou l'installation d'un nouveau compte système). Enfin, XMeta peut proposer les techniques investigations (**IT**) à utiliser, à partir d'une liste que nous avons établie (cf. TAB. 2).
La vérification d'hypothèses sur les attaques ou les actions complémentaires peut être indiquée (valeur du nœud correspondant positionnée à "observé") ; cette modification est propagée dans le réseau, qui peut proposer de nouvelles hypothèses.

A la fin de l'analyse d'un système (la source de **cette** attaque a été trouvée), il y a plusieurs possibilités :

1. la source de l'attaque est locale,

Nom de la variable	Description
image	création d'une image disque d'un média
syst_check	vérification des fichiers d'audit du système
net_check	vérification des fichiers d'audit réseau (pare-feu, etc.)
syst_var	vérification des variables système (login, processus, etc.)
retrieve	récupération des fichiers cachés ou supprimés
net_log	utilisation d'une scan réseau pour surveiller les actions de l'attaquant
int_topo	vérification de la topologie du réseau compromis
ext_topo	vérification des interconnexions du réseau compromis
comm	analyse des communications comme les logs IRC, les mails, etc.
physic	analyse des accès physiques à un système ou un bâtiment.

TAB. 2 – Techniques d'investigation

   a) soit l'attaquant avait un accès légitime au système, dans ce cas l'enquête technique est terminée,

   b) soit l'attaquant a illégitimement obtenu un accès avant de lancer son attaque ; dans ce cas l'enquête continue et un nouveau plan d'investigation est créé, avec la même configuration logicielle que précédemment et en prenant en compte comme dégât observé cette nouvelle information.

2. la source de l'attaque n'est pas locale (interne ou externe), dans ce cas la suite de l'enquête dépend de la capacité des enquêteurs à avoir accès au système en cause pour pouvoir poursuivre leur analyse.

Il est possible de créer autant de plans d'investigation que nécessaire. Des liens entre ces plans conservent la trace du cheminement de l'attaquant (des liens multiples sont autorisés).

3.2 Implémentation

L'implémentation actuelle est basée sur le moteur d'inférence de "Bayesian Tools in Java" [Hsu] et une interface utilisateur codée en Python/GTK. L'initialisation du moteur d'inférence devrait se faire grâce aux résultats d'enquêtes précédentes. Manquant de données réelles, nous avons dans un premier temps utilisé les seules informations contenues dans la base ICAT, à savoir les relations entre les logiciels, les dégâts. Une analyse lexicale de la base nous permet d'extraire les attaques et les actions complémentaires correspondantes. Nous obtenons ainsi une base de données utilisable par l'algorithme d'apprentissage *k2*. La première version de notre plateforme ne pouvait pas proposer de techniques d'investigation.

Prenons l'exemple (fictif) de données confidentielles ayant été volées sur un poste de travail. Le système compromis est un système Linux Debian, le plan d'investigation a été initialisé avec les logiciels d'une livraison Debian standard (kernel, libc, Windowmaker, OpenSSH, etc.) et un dégât de type *Confidentiality* (perte de confidentialité selon la dénomination ICAT). XMeta répond que la source de l'attaque a toutes les chances d'être locale (ce qui est vrai), et que les logiciels les plus vulnérables sont XFree86, la libc Linux et le noyau (kernel) (le composant attaqué est en réalité le noyau). Enfin, il s'avère que l'attaquant a utilisé un exploit pour devenir administrateur du système et copier les fichiers. Cette attaque se positionne en $7^{ème}$ position sur 21. Dans le cas d'une investigation réelle, toutes les attaques potentielles auraient dû être vérifiées, sachant que certaines d'entre elles peuvent facilement être écartées en fonction du contexte (par exemple, un DoS (*Denial of Service*) sur la machine contenant les données volées peut être dans ce cas écarté).

Après ce type de premiers essais, notre système a été enrichi de façon à pouvoir proposer en réponse les techniques d'investigation présentées dans TAB. 2.

4 L'affaire Kevin Mitnick

4.1 Introduction

Nous proposons ici de revisiter avec XMeta une partie de l'une des affaires les plus connues au monde, présentée en détail dans de nombreux articles, notamment par Tsutomu Shimomura, propriétaire des systèmes compromis [SM96]. L'affaire Mitnick est intéressante car c'est une attaque complexe mettant en

jeu plusieurs systèmes informatiques en parallèle. Tsutomu Shimomura est chercheur (*Senior Fellow*) au *Supercomputer Center* de San Diego, Californie, et travaille dans différents domaines de recherche, dont la sécurité informatique. En 1994, un inconnu s'infiltre dans les systèmes de Tsutomu et, après 7 semaines de traque intensive, Kevin Mitnick est localisé à Raleigh, Caroline du nord. Mitnick est arrêté le 14 février 1995, puis inculpé pour intrusion dans des systèmes d'information et vol de données confidentielles.

4.2 Notre enquête

Quand les ordinateurs de Tsutomu ont été attaqués, les premiers indices ont été trouvés sur l'un de ses systèmes, *Ariel* (voir Fig. 4). Beaucoup de scans réseaux avaient été effectués sur son réseau la nuit précédente. Un très gros fichier (*oki.tar.Z*) avait été créé, transféré vers une destination inconnue, puis supprimé du système. Il s'avéra par la suite que ce fichier contenait notamment des données confidentielles sur des logiciels de téléphones portables.

Ces premières informations ont été saisies dans XMeta (comme nous n'avons pas beaucoup de précisions sur les logiciels installés, leur liste peut être incomplète ou inexacte) :

> **ARIEL** :
> **Logiciels :** SunOS, GNU Tar, GNU Ghostscript, fingerd, ruserd et FTP
> **Dégats :** LT_Confidentiality

Le système XMeta répond :

> Les **attaques** les plus probables sont : contournement d'un équipement de sécurité (65%), diversion (masquage d'une attaque sur un autre système, 56%) et attaque par force brute (56%).
> Les **actions complémentaires** les plus probables sont : infection de fichiers (83%), inhibition de la détection (81%) et installation d'un nouveau compte système (71%).
> Les logiciels suspectés sont : GNU Tar (73%), Finger Service (73%) et FTP (27%). Aucune technique d'investigation n'est proposée, ce qui signifie qu'il n'y a pas de cas similaire dans notre base de données des enquêtes précédentes.

Les éventuels fichiers d'audit des applications suspectées devraient donc être vérifiés. Les 3 attaques proposées laissent à supposer que l'action devait venir de l'extérieur, cela étant confirmé par le nombre important de scans réseaux observés par Tsutomu. L'attaquant a contourné un équipement (ou un service) de sécurité ou bien il a fait une diversion pour s'emparer du fichier *oki.tar.Z*. On peut ignorer l'attaque par force brute car elle ne permet pas de s'emparer d'un fichier. Cependant, la présence de cette attaque peut s'expliquer facilement : étant donné le peu d'expérience accumulée, notre base de connaissance est le reflet des vulnérabilités répertoriées dans la base ICAT. Les probabilités estimées sont fonction de la fréquence d'apparition des logiciels et des attaques dans cette base. Ce disfonctionnement devrait disparaître au fûr et à mesure de l'enrichissement de la base d'expérience.

Tsutomu a trouvé que tous les scans réseaux provenaient du domaine Internet *toad.com* (FIG. 5 extraite de [Shi]). Il a aussi trouvé que son ordinateur nommé *Osiris* avait un comportement étrange, avec des fenêtres vides affichées en haut de l'écran. Nous avons donc les faits suivants :

- *Ariel* et *Osiris* ont des relations de confiance fortes,
- *Osiris* est physiquement au domicile de Tsutomu, qui n'a pas de relation avec *toad.com*,
- beaucoup de trafic réseau a été observé vers *Osiris* (FIG. 6).

Tout ceci implique que l'enquête doit continuer sur *Osiris* et pas (dans un premier temps) sur le réseau de *toad.com*. *Osiris* peut-être la source de l'attaque ou au moins un vecteur. Un nouveau plan d'investigation est donc crée sur *Osiris*. Tsutomu a découvert que cet ordinateur semblait être déconnecté du réseau de son bureau et en particulier d'*Ariel*.

> **OSIRIS** :
> **Logiciels :** SunOS, GNU Tar, GNU Ghostscript, fingerd, ruserd et FTP
> **Dégats :** : LT_Availability

Le système XMeta répond :

> Les **attaques** les plus probables sont : repeat (*scanning sweeping* par exemple, 100%), overrun (DOS, DDOS, smurf, fraggle, etc., 89%) et bypass (contournement d'un équipement de sécurité, 68%)
> Les **actions complémentaires** sont : infection (73%), trap (installation d'une backdoor, 62%) et del (suppression de données, 45%)
> Les logiciels potentiellement cibles sont : FTP (73%), GNU Tar (38%) et GNU Ghostscript (38%).

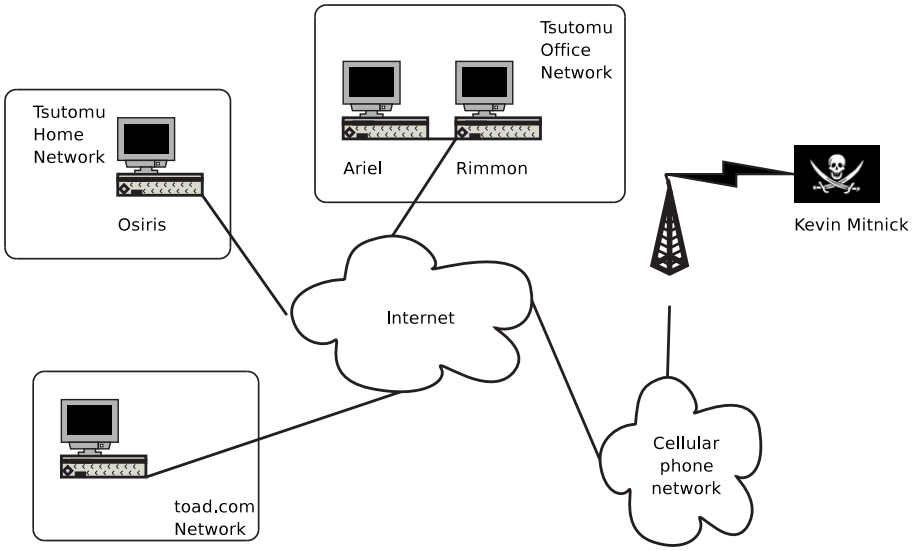

FIG. 4 – Les ordinateurs impliqués dans l'attaque

```
14:09:32 toad.com# finger -l @target
14:10:21 toad.com# finger -l @server
14:10:50 toad.com# finger -l root@server
14:11:07 toad.com# finger -l @x-terminal
14:11:38 toad.com# showmount -e x-terminal
14:11:49 toad.com# rpcinfo -p x-terminal
14:12:05 toad.com# finger -l root@x-terminal
```

FIG. 5 – Premiers logs de *toad.com* ; target correspond à *Ariel*, server correspond à *Rimmon* et x-terminal correspond à *Osiris*

Cela implique qu'*Osiris* n'est peut-être qu'un vecteur pour l'attaquant, car ce dernier n'a pas pu pénétrer dans le système par des scans réseau ou par déni de service. Ces résultats nous incitent à explorer une autre piste.

Osiris est un Terminal X-Window connecté à *Rimmon*, qui a donc pu être aussi attaqué. Ceci est confirmé par les logs récupérés par Tsutomu (FIG. 7). Comme nous n'avons pas beaucoup d'informations sur *Rimmon*, nous supposons qu'il est configuré comme *Osiris* et *Ariel*. Tsutomu a découvert qu'un utilisateur non autorisé a réussi à installer un module-noyau nommé *Tap 2.01* sur *Rimmon*, ce qui implique d'avoir des droits administrateur sur la machine (FIG. 8).

RIMMON :
Logiciels : SunOS, GNU Tar, GNU Ghostscript, fingerd, ruserd et FTP
Dégâts : LT_Obtain_all_priv et LT_Availability

XMeta répond :

Les **attaques** les plus probables sont : Trojan (installation d'un cheval de Troie, 93%), bypass (78%) et force brute (58%).
Les **actions complémentaires** les plus probables sont : login_inst (installation d'un compte, 58%), infection de fichier (51%) et trap (46%)
Les logiciels potentiellement cibles : FTP (59%) et GNU Tar (41%).

Grâce à ces résultats, si nous ne trouvons pas de cheval de Troie sur *Rimmon*, cela signifie que ce dernier a aussi été utilisé comme vecteur d'attaque comme *Osiris*. Tsutomu n'a pas trouvé de cheval de Troie, mais une attaque par *Flooding* (appelée *Overrun* dans XMeta). Nous pouvons maintenant affirmer que *Rimmon* était bien un moyen d'accéder à *Osiris* et *Ariel*. L'attaque *Overrun* n'était classée qu'en $10^{ème}$ position[1] sur 21. Mais Tsutomu a aussi trouvé que l'attaquant a installé un module dans le noyau, il avait donc les droits admi-

[1] de 1 à 10 : Trojan, bypass, brute_force, broadcast, chaff, repeat, intercept, net_listen, bounce et overrun

```
14:18:25.906002 apollo.it.luc.edu.1000 > x-terminal.shell: S 1382726990:1382726990(0) win 4096
14:18:26.094731 x-terminal.shell > apollo.it.luc.edu.1000: S 2021824000:2021824000(0) ack 1382726991 win 4096
14:18:26.172394 apollo.it.luc.edu.1000 > x-terminal.shell: R 1382726991:1382726991(0) win 0
14:18:26.507560 apollo.it.luc.edu.999 > x-terminal.shell: S 1382726991:1382726991(0) win 4096
14:18:26.694691 x-terminal.shell > apollo.it.luc.edu.999: S 2021952000:2021952000(0) ack 1382726992 win 4096
14:18:26.775037 apollo.it.luc.edu.999 > x-terminal.shell: R 1382726992:1382726992(0) win 0
...
```

FIG. 6 – logs provenant d'*Osiris*

```
14:18:22.516699 130.92.6.97.600 > server.login: S 1382726960:1382726960(0) win 4096
14:18:22.566069 130.92.6.97.601 > server.login: S 1382726961:1382726961(0) win 4096
14:18:22.744477 130.92.6.97.602 > server.login: S 1382726962:1382726962(0) win 4096
14:18:22.830111 130.92.6.97.603 > server.login: S 1382726963:1382726963(0) win 4096
14:18:22.886128 130.92.6.97.604 > server.login: S 1382726964:1382726964(0) win 4096
...
```

FIG. 7 – logs provenant de *Rimmon*

nistrateur.

4.3 Synthèse des résultats obtenus avec XMeta

Nous avons trouvé les éléments suivants :

- un fichier *oki.tar.Z* a été transféré d'*Ariel* vers une destination inconnue en utilisant soit un contournement soit une diversion,
- un des ordinateurs de *toad.com* a été utilisé pour analyser le réseau de Tsutomu et ses relations de confiance,
- sur *Osiris*, l'attaquant a utilisé soit l'attaque *repeat* pour obtenir des informations, ou bien *Bypass* pour pénétrer dans le système,
- l'attaquant a utilisé les relations de confiance entre *Ariel* et *Osiris* pour accéder à *Ariel*,
- l'attaquant a utilisé les relations de confiance entre *Osiris* et *Rimmon* pour accéder à *Osiris*,
- l'attaquant a été capable d'installer un module noyau sur *Osiris* et ce module a certainement été utilisé pour accéder à *Ariel*.

Ces résultats peuvent être sauvegardés pour des enquêtes futures, et complétés en précisant les techniques employées pour analyser chacun des systèmes. La sauvegarde permet aussi par la suite de "rejouer" l'attaque ou l'enquête, par exemple à des fins d'explication devant un tribunal. Cette sauvegarde se fait actuellement en XML, dans le format CFXR, *Computer Forensics XML Report*, que nous avons défini.

Dans le cadre de l'enquête présentée ici, les prochaines étapes seraient de comprendre comment l'attaquant a pu obtenir les droits administrateur sur *Osiris* et sur les ordinateurs de *toad.com* puis d'essayer d'identifier la destination du fichier *oki.tar.Z*.

4.4 Les conclusions de l'enquête de Tsutomu Shimomura

L'attaque décrite par Tsutomu Shimomura s'est jouée en trois temps.

Dans une première phase, l'attaquant a essayé de prédire les numéros de séquence initials des connexions TCP acceptés par *Osiris* en lui envoyant un nombre important de paquets TCP SYN (cette attaque est appelée *Repeat* dans XMeta), puis en les annulant avec des paquets TCP RST (Fig. 9).

Il a pu ensuite envoyer à *Osiris* par *rsh* la commande "`echo ++ >>/.rhosts`" en se faisant passer pour *Rimmon* (Fig. 10). Il a ainsi obtenu un accès total à cette machine. Cette attaque est nommé *contournement d'un équipement de sécurité* dans XMeta. Nous avons vu qu'elle était classée en troisième position avec une probabilité de 68%. L'attaque par *Flooding* (*Overrun* dans XMeta) a été utilisée pour "baillonner" *Rimmon* et l'empêcher de répondre à *Osiris* (Fig. 10).

Dans la troisième phase de l'attaque, l'attaquant a installé le programme *Tap* sur *Osiris*, ce qui lui a permis de prendre le contrôle de la relation de confiance entre *Osiris* et *Ariel* (Fig. 11). Il a eu alors accès à *Ariel* et a pu créer, transférer puis supprimer le fichier *oki.tar.Z*.

Conclusion et travaux futurs

Nous proposons un système expert d'assistance aux enquêtes criminelles informatiques. Pour une configuration et un type d'accident donnés, XMeta propose des hypothèses sur les logiciels les plus vulnérables et la manière de procéder de l'attaquant. Des indications sur les techniques d'investigation à employer peuvent aussi être fournies. L'application à une partie de l'affaire Kevin Mitnick montre que notre système peut traiter des

```
x-terminal% modstat
Id  Type  Loadaddr      Size   B-major  C-major  Sysnum   Mod Name
 1  Pdrv  ff050000      1000             59.            tap/tap-2.01 alpha

x-terminal% ls -l /dev/tap
crwxrwxrwx  1 root      37,  59 Dec 25 14:40 /dev/tap
```

FIG. 8 – Quelques variables système de *Rimmon*

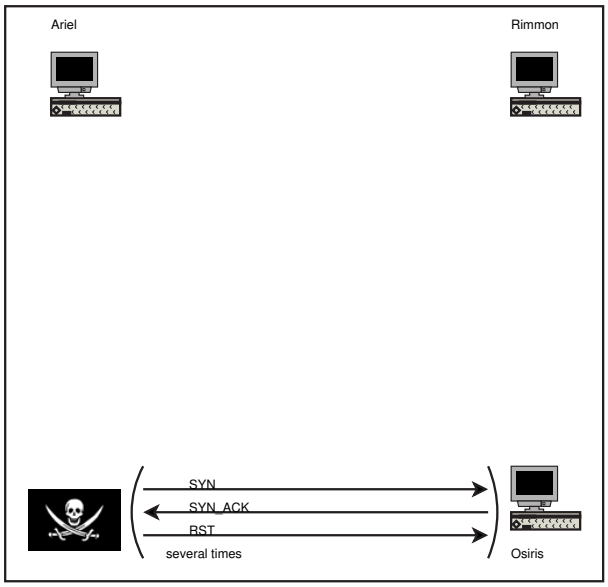

FIG. 9 – l'attaque de Kevin Mitnick : première étape

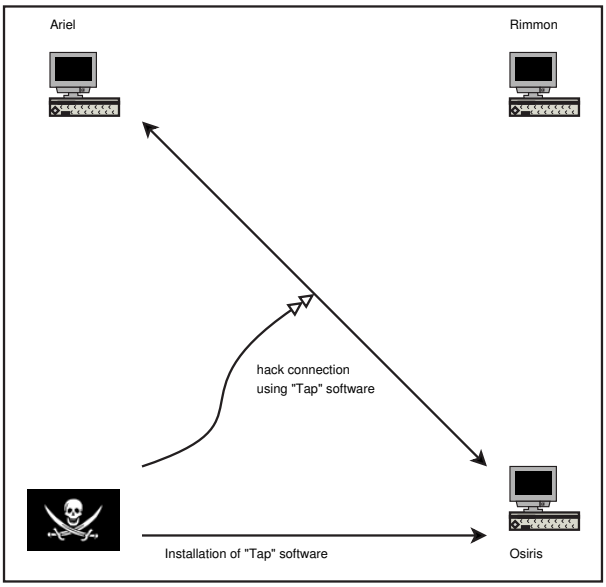

FIG. 11 – l'attaque de Kevin Mitnick : troisième étape

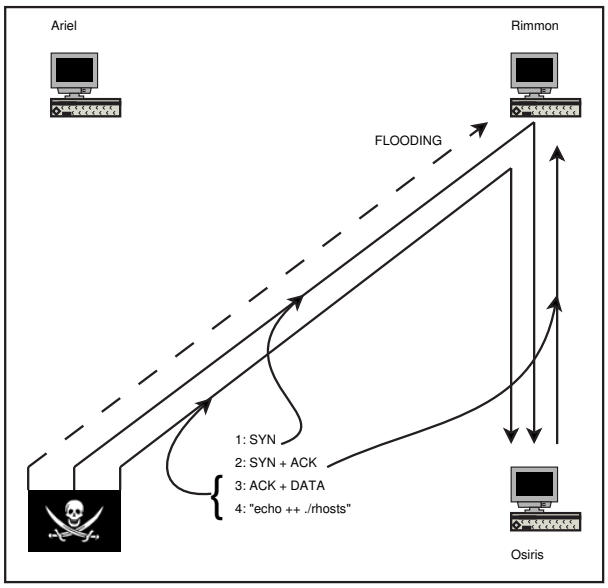

FIG. 10 – l'attaque de Kevin Mitnick : deuxième étape

cas concrets et non triviaux. Des efforts doivent encore être faits pour enrichir notre base d'expérience sur les enquêtes précédentes (en particulier pour les techniques d'investigation). XMeta fournit non seulement une assistance en cours d'enquête, mais permet aussi de sauvegarder des résultats réutilisables sur le cheminement de l'attaque comme de l'enquête.

Nous travaillons maintenant sur la notion de profil d'attaquant, l'objectif étant d'orienter l'enquête en fonction des habitudes des délinquants (logiciels cibles préférés, attaques préférées). Nous souhaitons aussi apporter une dimension plus dynamique à XMeta, en y incluant des informations sur la chronologie des actions élémentaires réalisées par un attaquant pour atteindre son objectif.

Références

[BWC02] D. J. Burroughs, L. F. Wilson, and G. V. Cybenko. Analysis of distributed intrusion

detection systems using bayesian methods. In *IPCCC 2002*, April 2002.

[CH92] G. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9 :309–347, 1992.

[Chr02] Alan M. Christie. The incident detection, analysis, and response (idar) project. Technical report, CERT Coordination Center, July 2002. http ://www.cert.org/idar.

[CJLM03] Paulo Costa, Jim Jones, Billy Liao, and Vijay Malgari. A system for collection, storage, and analysis of multi-platform computer system data. Technical report, George Mason University, 2003.

[CLU] CLUSIF. Club de la sécurité des systèmes d'information français. https ://www.clusif.asso.fr/.

[FC89] R. Fung and K.C. Chang. Weighting and integrating evidence for stochastic simulation in bayesian network. In *Proc. of UAI*, volume 5, pages 209–219. New York Elsevier science publishing company, Inc, 1989.

[Hsu] William H. Hsu. BNJ - Bayesian Network tools in Java. http ://bndev.sourceforge.net.

[LL00] Tod S. Levitt and Kathryn Blackmond Laskey. Computational inference for evidential reasoning in support of judicial proof. In *Symposium - Artificial Intelligence and Judiciary Proofs*, 2000.

[MP01] Kevin Mandia and Chris Prosise. *Incident Response : Investigating Computer Crime*. McGraw-Hill Osborne Media, June 2001. ISBN : 0072131829.

[NWL+04] Patrick Naïm, Pierre-Henri Wuillemin, Philippe Leray, Olivier Pourret, and Anna Becker. *Réseaux bayésiens*. Eyrolles, 2004.

[Pop00] Sébastien Populaire. Introduction aux réseaux bayésiens. Technical report, Heudiasyc, Université de Technologie de Compiègne (France), 2000.

[Shi] Tsutomu Shimomura. Technical details of the attack described by markoff in nyt. Article : 14059 of comp.security.misc. 25 Jan 1995 04 :36 :37 -0800.

[SM96] Tsutomu Shimomura and John Markov. *Takedown*. New York : Hyperion Press, 1996.

[teaa] ICAT team. ICAT vulnerability database . http ://icat.nist.gov/icat.cfm.

[Teab] SpamAssassin Team. Spamassassin. http ://spamassassin.apache.org. An extensible email filter used to identify spam.

Annexe : Les variables d'attaque et d'action

Nom de la variable	Description
listing	lister des entrées DNS par exemple
net_listen	écouter le réseau
decrypt	utiliser une attaque par dictionnaire ou par force brute pour déchiffrer un message
exploit	utiliser un exploit (Buffer Overflow)
bypass	contourner un élément de sécurité
broadcast	trouver un système en envoyant des paquets en diffusion
chaff	utiliser un faux serveur pour voler des données
embezzlement	exemple : Man In the Middle
listen	écouter les évènements d'un système
parasit	transformer les fonctionnalités d'un logiciel
degrade	dégrader le service d'un système (exemple : défigurement d'une page web)
diversion	utiliser une diversion
intercept	intercepter des données qui étaient destinées à quelqu'un d'autre
usurp	utiliser l'identité de quelqu'un sans son accord
bounce	rebondir sur plusieurs systèmes avant d'attaquer
troyen	utiliser un cheval de troie
repeat	par exemple : scanning sweeping
blocking	bloquer les functionnalités d'un service réseau
overrun	saturation comme par exemple : DoS, DDoS
brut_force	attaquer par force brute
control	intercepter et bloquer les communications d'un système

TAB. 3 – Variables d'attaque

Nom de la variable	Description
msg	envoyer un message pour signer l'attaque
attribute	augmentation des privilèges
scan_use	trouver les services d'un système par des scans réseau
encrypt	chiffrer ses données
hidden_channel	utiliser une faiblesse dans un protocole pour envoyer des données
infection	ajouter des informations cachées dans un fichier (exemple : stéganographie)
cnx_illic	se connecter illicitement à un système
trap	utiliser une porte cachée
invert_trap	utiliser une porte cachée inversée
inhib_detect	inhibition de la detection (exemple : IP Spoofing)
del	supprimer des données
login_inst	installer un nouveau compte utilisateur

TAB. 4 – variables d'action