

100 % SÉCURITÉ INFORMATIQUE



MISC

Multi-System & Internet Security Cookbook

HORS - SERIE

France METRO : 9 €
DOM : 9 €
CAN : 13,50 \$CAD
CH : 15 CHF
BEL : 9,90 €
POL/S : 1100 CFP
POL/A : 1400 CFP

L 16844 - 10 H - F: 9,00 € - RD



OCTOBRE
NOVEMBRE
2014 **N°10**

INTRODUCTION

- Qu'est-ce qu'un incident de sécurité ?
- Aspects juridiques du traitement d'incident

ANALYSE TECHNIQUE

- Recherche d'indices de compromission sur iOS
- Réponse à incident et forensic sur Android
- Méthodologie d'analyse de vulnérabilités
- Vue globale d'une analyse de malware

RÉAGIR À UNE INTRUSION

LA CITADELLE EST TOMBÉE...



...QUE DOIS-JE FAIRE ?

TRAITER LES INCIDENTS À GRANDE ÉCHELLE

- La citadelle est tombée ! Cybercrise, comment s'organiser ?
- Armez-vous pour analyser les journaux et le réseau
- Mais où est Charlie ou comment débusquer un attaquant dans un parc de machines ?
- Analyse pratique d'un Active Directory avec BTA

NE MANQUEZ PAS MISC N°75 !

DOSSIER SPÉCIAL :

LES OBJETS CONNECTÉS



**DE NOUVEAUX
USAGES
PARTOUT,
LA SÉCURITÉ
NULLE PART ?**

**DISPONIBLE CHEZ VOTRE MARCHAND
DE JOURNAUX ET SUR :
boutique.ed-diamond.com**



APTE OU IN-APT, FACE AUX A.P.T. ? À VOUS DE VOIR !

La mode des APT est bien implantée depuis maintenant quelques années et l'actualité sécu est rythmée par la publication des rapports de sociétés soucieuses de surfer sur le buzz des noms les plus improbables (vivement le rapport « tardigrade irradié » !).

Le nombre de ces rapports montre que, heureusement, les victimes commencent à découvrir ces attaques et parfois même à les rechercher. En France, la médiatisation de l'attaque de Bercy en mars 2011 a sûrement contribué à réveiller les esprits les plus endormis.

La réponse aux incidents est donc hype, utilisée commercialement pour se proclamer expert, capable de détecter, corriger et prévenir les attaques à grand renfort de « spectrométries sur équipements » ou autre « cyber intelligence » ! Mais au final, combien d'équipes sont déjà réellement intervenues sur des attaques de grande ampleur et donc aptes à fournir le service promis ? Car le forensics à l'ancienne a vécu et personne ne peut plus traiter l'incident seul à coups de *dd* et de *grep*. Aujourd'hui, l'APT ne peut être considérée que comme une crise où, nécessairement, la dimension humaine est cruciale. De plus, l'ensemble des domaines techniques de la SSI est concerné ; au-delà des OS traditionnels, tous les composants du SI : smartphones, IPBX, cœur de réseau, équipements industriels, etc. font partie du périmètre à analyser.

Bref, la situation est dramatique dans un monde qui se divise en deux catégories : ceux qui connaissent leur compromission et ceux qui doivent encore la découvrir. Or, celle-ci est encore vue comme l'étaient les poux : une maladie honteuse. Pourtant, seule la collaboration permettra de combattre l'épidémie et de réduire l'asymétrie entre attaquant et défenseur. Pour lutter efficacement, il est nécessaire de mettre en place un partage entre tous les acteurs sur les outils, les techniques d'attaques, les marqueurs, etc. Reste à voir d'où viendra le salut. Peut-être l'ANSSI donnera-t-elle (enfin !) l'impulsion du partage généralisé ? Ou faudra-t-il attendre un acteur privé de référence en Europe continentale, à l'instar de FireEye/Mandiant ou BAE/Detica ?

En attendant cette révolution, il faut jouer les pompiers et j'espère que ce numéro vous donnera non seulement un aperçu des difficultés auxquelles il faut faire face, mais également quelques idées à mettre en œuvre.

Raphaël Rigo



www.miscmag.com

MISC est édité par Les Éditions Diamond
B.P. 20142 / 67603 Sélestat Cedex
Tél. : 03 67 10 00 20 - Fax : 03 67 10 00 21
E-mail : cial@ed-diamond.com
Service commercial : abo@ed-diamond.com
Sites : www.miscmag.com
boutique.ed-diamond.com
IMPRIMÉ en Allemagne - PRINTED in Germany
Dépôt légal : A parution
N° ISSN : 1631-9036
Commission Paritaire : K 81190
Périodicité : Bimestrielle
Prix de vente : 9 Euros

Directeur de publication : Arnaud Metzler
Chef des rédactions : Denis Bodor
Rédacteur en chef : Frédéric Raynal
Secrétaire de rédaction : Aline Hof
Conception graphique : Kathrin Scall
Responsable publicité :
Black Mouse Communication - Tél. : 03 67 10 00 27
Service abonnement : Tél. : 03 67 10 00 20
Impression : pva, Druck und Medien-Dienstleistungen GmbH, Landau, Allemagne
Illustrations : www.fotolia.com
Distribution France : (uniquement pour les dépositaires de presse)
MLP Réassort :
Plate-forme de Saint-Barthélemy-d'Anjou. Tél. : 02 41 27 53 12
Plate-forme de Saint-Quentin-Fallavier. Tél. : 04 74 82 63 04
Service des ventes : Distri-médias : Tél. : 05 34 52 34 01



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 Les Éditions Diamond. 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.



Charte de MISC

MISC est un magazine consacré à la sécurité informatique sous tous ses aspects (comme le système, le réseau ou encore la programmation) et où les perspectives techniques et scientifiques occupent une place prépondérante. Toutefois, les questions connexes (modalités juridiques, menaces informationnelles) sont également considérées, ce qui fait de MISC une revue capable d'appréhender la complexité croissante des systèmes d'information, et les problèmes de sécurité qui l'accompagnent. MISC vise un large public de personnes souhaitant élargir ses connaissances en se tenant informées des dernières techniques et des outils utilisés afin de mettre en place une défense adéquate. MISC propose des articles complets et pédagogiques afin d'anticiper au mieux les risques liés au piratage et les solutions pour y remédier, présentant pour cela des techniques offensives autant que défensives, leurs avantages et leurs limites, des facettes indissociables pour considérer tous les enjeux de la sécurité informatique.

INTRODUCTION

- [04] QU'EST-CE QU'UN INCIDENT DE SÉCURITÉ ?
- [11] ASPECTS JURIDIQUES ET LÉGAUX DU TRAITEMENT D'INCIDENT

ANALYSE TECHNIQUE « FORENSICS »

- [18] ANALYSE DE MALWARE À LA RESCOURS DU CSIRT : DE LA RÉTRO-CONCEPTION AUX IOC
- [25] MÉTHODOLOGIE D'ANALYSE DE VULNÉRABILITÉS ET PRODUCTION DE SIGNATURES DE DÉTECTION
- [33] RECHERCHE D'INDICES DE COMPROMISSION SUR IOS
- [45] QUELLE MÉTHODOLOGIE POUR UNE RÉPONSE À INCIDENT SUR ANDROID ?

TRAITER UN INCIDENT À GRANDE ÉCHELLE

- [54] LA CITADELLE EST TOMBÉE ! CYBERCRISE, COMMENT S'ORGANISER ?
- [61] DÉBUSQUER L'APT DANS VOTRE PARC
- [68] INTRODUCTION AU TRAITEMENT DES LOGS ET DES FLUX RÉSEAU DANS UNE RÉPONSE SUR INCIDENT
- [75] CONTRÔLER LA SÉCURITÉ DES OBJETS DE L'ACTIVE DIRECTORY AVEC BTA

ABONNEMENT

- [31 / 32] BONS D'ABONNEMENT ET DE COMMANDE

SUIVEZ LES DERNIÈRES ACTUALITÉS DE VOTRE MAGAZINE SUR :

- FACEBOOK :**
<https://www.facebook.com/editionsdiamond>
- TWITTER :**
<https://twitter.com/miscredac>



DÉFINIR « L'INCIDENT DE SÉCURITÉ »

Vincent Martin – vmartin@silent.works

mots-clés : INCIDENT / ÉVÉNEMENT / RISQUE / ADAPTATION / CERT/CSIRT

I SO 27035, page 7, paragraphes 4 et 5. Merci, bonne journée.
Qu'il serait pratique de pouvoir s'arrêter à la définition de l'ISO de « l'incident de sécurité ». L'auteur n'aurait pas à se fatiguer davantage à écrire et, cher (chère) lecteur (trice), avec les idées enfin claires, nous pourrions passer à ce pour quoi nous tenons réellement ce recueil entre nos mains : les articles techniques.

Pourtant, la question mérite sans doute d'être posée, car le gestionnaire de la sécurité vit dans la peur de cet incident de sécurité et ce même incident est ce qui justifie l'existence d'investigateurs, d'équipes de CERT, CSIRT et autres « analystes forensic ». C'est, pour utiliser les termes à la mode, l'élément précurseur de tout APT qui se respecte et définir convenablement un incident de sécurité est ce qui permet de lancer proprement la démarche de réponse sur incident.

Mais comment définit-on ce qu'est un incident ? En fonction de ce qu'est un événement ? En fonction de ses effets sur un SI ? Une définition immuable dans le temps et dans l'espace est-elle envisageable ?

Poser le problème permet d'entrevoir des pistes de réflexion face à une question moins simple qu'il n'y paraît. On s'efforcera de répondre en trois parties :

- Pourquoi une approche figée de la définition de l'incident de sécurité atteint-elle ses limites dans l'état actuel de la sécurité ?
- Quel type de démarche permet une approche flexible, plus à même de fournir des éléments « exploitables » pour ceux en charge de l'investigation ?
- En quoi une définition flexible de l'incident de sécurité nous donne des pistes pour améliorer globalement la sécurité des organisations ?

1 Une approche « sortie des livres » finit aujourd'hui par atteindre ses limites

Commençons par poser quelques questions, comme ça, juste pour voir :

- Comment une organisation sait-elle qu'elle fait face à un incident de sécurité ?
- Dans une entreprise ou une organisation, si le RSSI dresse une liste *a priori* de ce que sont « les incidents de sécurité », cette liste reste-t-elle immuable ? Sinon, quand et pourquoi la faire évoluer ?
- Si vous prenez deux entités distinctes (appartenant ou pas à la même organisation), ces dernières sont-elles censées adopter une liste identique d'« incidents de sécurité » ? Si non, qu'est-ce qui justifierait une différence entre les approches ?

Poser ces trois questions doit permettre d'entrevoir, derrière l'apparente simplicité d'une taxonomie (« Donne-moi des critères et je te dirai ce qu'est un incident »), qu'une façon moins périlleuse de répondre à la question est de faire preuve de méthode et de pragmatisme, à savoir : « s'appuyer sur la pratique ».



1.1 Où l'on pourrait se raccrocher à l'investigation policière, ou pas...

Commençons par un détour par la criminalistique. Au XVIII^e siècle, le pénaliste italien Cesare Beccaria accouche d'une formule devenue célèbre : « *Nullum crimen, nulla poena sine lege* » qui dispose, selon Wikipédia, qu'on ne peut être condamné pénalement qu'en vertu d'un texte pénal précis et clair. C'est le principe de légalité des délits et des peines en droit pénal.

Ceci est bien beau me direz-vous, mais quel est le lien avec notre problème ? Pour répondre, considérons le métier d'enquêteur de police. En matière pénale, on enquête en fonction d'une incrimination possible, en fonction de ce qui est contenu dans le Code pénal. En somme, on a une liste de ce qui est un trouble à l'ordre public, plus ou moins grave, et il faut établir ce qui, dans les agissements d'un individu, est constitutif d'un trouble défini dans le Code. Pour prendre un exemple qui devrait être bien connu dans nos professions, le Code pénal, dans son article 323-1 dispose que : « *Le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni [...]. Lorsqu'il en est résulté soit la suppression ou la modification de données contenues dans le système, soit une altération du fonctionnement de ce système [...]. Lorsque les infractions prévues aux deux premiers alinéas ont été commises à l'encontre d'un système de traitement automatisé de données à caractère personnel mis en œuvre par l'État [...]* ». L'investigateur doit donc apporter au juge les éléments tangibles qui permettront de caractériser l'intrusion, l'altération de fonctionnement, le caractère frauduleux, etc.

Dans ce cas, nous avons donc une liste de tout ce qui constitue des délits et des crimes et le Code pénal (spécial) donnera à l'enquêteur l'ensemble des éléments constitutifs à la chasse desquels il pourra se lancer pour assurer l'inculpation et la condamnation éventuelle.

Le Code pénal est donc organisé en fonction de la gravité du trouble à l'ordre public (ce qui est « mal »), et comme les sociétés évoluent lentement, ces troubles évoluent peu, et la liste de ce qui est répréhensible ne bouge pas beaucoup.

La question est donc, si on veut transposer cette méthode au monde des SI, de déterminer si l'on peut avoir un « Code pénal » unique, et donc s'il existe un trouble à l'ordre public des SI, immuable.

1.2 Les définitions de l'ISO

Dans les SI, il n'y a pas beaucoup de lois, mais il y a des palanquées d'ISOs et autres RFCs. Au sein des ISOs, la série des 27xxx traite de la sécurité des

systèmes d'information. La 27035 traite de la gestion des incidents de sécurité. On pourrait donc parier que la définition des incidents de sécurité permettrait, une bonne fois pour toutes, de définir ce qu'est un « trouble à l'ordre public informatique ou informationnel ».

Événement de sécurité (définition ISO) :

« *Occurrence identifiée de l'état d'un service, d'un système ou d'un réseau indiquant une faille possible dans la politique de sécurité de l'information ou un échec des mesures de sécurité ou encore une situation inconnue jusqu'alors et pouvant relever de la sécurité* ».

Incident de sécurité (définition ISO) :

« *Un ou plusieurs événements liés à la sécurité de l'information indésirables ou inattendus présentant une probabilité forte de compromettre les activités de l'organisation et de menacer la sécurité de l'information* ».

L'ISO explicite certains critères : un incident de sécurité peut être « *délibéré* » ou « *accidentel* », causé par des « *moyens techniques ou physiques* ».

Leurs conséquences peuvent être la fuite, modification, destruction ou indisposition d'information par des moyens non autorisés, ou le fait d'endommager ou voler des biens de l'organisation. L'ISO n'oublie pas de mentionner que le fait de ne pas relever ou rapporter un événement de sécurité peut avoir des conséquences sur la capacité d'une organisation à en contrer les effets et prévenir les réoccurrences.

Ces définitions sont extrêmement générales et nécessitent, pour être un tant soit peu gérables, de définir, dans chaque cas particulier, une liste d'items de sécurité qui seraient soit des *événements* de sécurité, soit des *incidents* de sécurité.

1.3 Faire une liste a priori des incidents de sécurité

Nous disposons donc d'une définition terriblement large de ce qui est un trouble, reste alors au responsable de la SSI de dresser sa propre liste de ce qui n'est « pas bien », et en avant pour le catalogue. Et ne nous mentons pas, cette approche est très, très séduisante. On peut se mettre derrière un clavier et commencer à taper au kilomètre une politique de sécurité. Le syndrome décrit ici est celui du tableur Excel avec des centaines d'entrées et en face les moyens de faire face à un incident identifié.

Petit problème : la mise en œuvre de cette politique sera-t-elle réaliste ? Correspondra-t-elle à la réalité opérationnelle ? Pourra-t-elle être appliquée par les personnels en charge de la SSI de l'organisation ? Plus difficile encore, Sécurité des SI et administration des SI doivent fonctionner ensemble, mais à un moment une PSSI intrinsèquement imparfaite ne va-t-elle pas compliquer la tâche de chacun ? Ici le problème



rencontré est : « tout sauf un audit » ! Vous êtes responsable de la sécurité de votre SI, vous avez vos firewalls (aucun de ces derniers n'est en ANY ANY, n'est-ce pas), vos proxies loguant et filtrant, etc. Quelle est votre préoccupation ? Que surtout rien ne bouge, tout cela repose sur un équilibre qui a plus à voir avec la magie noire qu'avec l'informatique.

La deuxième vague de problèmes apparaîtra invariablement en même temps qu'un problème indissociable de la nature même d'une crise de SSI : sauf exception, au début d'un incident on ne sait pas forcément ce à quoi l'on a affaire. Ce qui peut sembler trivial, sans importance particulière avant l'examen, se révélera par la suite avoir été le point de départ de la crise. Le fait que la secrétaire de Monsieur Duval, le DRH, a reçu la 1254ème présentation de chatons de l'année est-il le vecteur d'installation d'un RAT ? Une communication vers Dropbox® est-elle une exfiltration ?

Une approche à partir de l'ISO, ou de la théorie en général, est un peu inconfortable, car les livres ne nous disent pas vraiment comment faire pour définir un incident **dans sa propre organisation, concrètement, et dans sa propre chaîne de gestion de la sécurité.**

1.4 Concrètement, mettre en place une liste statique peut endormir une organisation dans un faux sentiment de sécurité

Un schéma classique fait au départ intervenir une liste dressée par le RSSI. Ce dernier met en place des moyens pour contrer ces événements (les « événements » ou « scénarios » redoutés), et il se sent libre de se désintéresser de la question.

Malheureusement, la menace évolue avec le temps qui passe, et les moyens mis en place hier peuvent se révéler largement insuffisants aujourd'hui et placer une organisation dans une situation critique demain.

En fait, l'approche « statique » a prouvé ses limites dans la pratique. Pour qui travaille aujourd'hui dans la protection des SI ou les investigations en cas d'incidents, notamment en ce qui concerne les fameuses APT, il est un constat assez simple à faire : il n'est pas franchement compliqué pour un attaquant de compromettre un système connecté à Internet. Pire encore, aujourd'hui on risque en particulier de ne pas voir que l'on est compromis. En effet, une grande entreprise peut perdre un contrat à l'exportation sans comprendre pourquoi, jusqu'à ce jour où, convoqué à un ministère, le directeur commercial et celui de la sécurité sont accueillis par une personne en costume (et parfois même cravate) qui leur met sous les yeux une copie de l'offre soumise par la société accompagnée de cette phrase devenue

rituelle ces dernières années : « *devinez où on a retrouvé ce document ? Pas là où il était censé apparaître* ».

Attention, il ne s'agit pas d'avancer qu'un fonctionnement à partir d'une liste prédéterminée d'incidents est une approche à bannir. Non, mais il faut la rapprocher des limites du Code pénal : c'est une approche nécessaire pour faire face aux délits et crimes connus, dans un contexte où la criminalité évolue peu. On se blindera contre toutes les attaques connues, sous réserve de ne pas en occulter et on fera en sorte que les risques « classiques » soient couverts, à un moment donné.

1.5 De la sécurité absolue à la sécurité relative

Bruce Schneier (oui, oui, le type qui a compté deux fois jusqu'à l'infini) faisait une remarque intéressante en juin dernier : « *Beaucoup de gens pensent que la sécurité est une affaire relative : pourvu que vous ayez les moyens de sécurité qui vous placent en haut à droite dans les diagrammes en quatre carrés, un attaquant ira voir votre voisin plutôt que de perdre son temps sur votre super défense faite de boîtiers d'analyses de malware, NIDS/NIPS, HIPS, SOCs, firewalls nouvelle génération et comportementaux, etc. Et c'est là que survient le drame... Car cela rend aveugle à l'attaquant vraiment dangereux : celui qui est déterminé, qui a du temps et de la ressource. Celui-là, il finira par rentrer, quoi qu'il arrive. La sécurité n'est plus alors un problème relatif : être plus difficile à atteindre qu'une autre cible potentielle, mais absolue, car il faut faire face à quelqu'un qui s'arme spécifiquement contre vous* ».

Il faut donc réfléchir à une approche d'un genre un peu nouveau : l'incident de sécurité ne se définirait plus alors comme un événement simplement adverse. Une approche « *known bad* », parce qu'elle requiert un référentiel externe, est limitée. Il faut trouver le moyen de ne pas simplement chercher à faire face à une attaque dans son type de contexte : gouvernemental, PME, banque, entreprise du CAC40. Il faut se détacher de l'approche relative de la sécurité, car elle ne protège en rien contre l'attaquant déterminé. Il faut réussir à déterminer ce qui est mauvais dans son *propre* contexte, en fonction de ce qui précieux pour *son* entité.

En ce sens, une grande entreprise industrielle française a eu un réflexe intéressant il y a déjà une petite dizaine d'années. Sa direction, voyant qu'elle détectait de moins en moins d'attaques s'est retrouvée devant la question : soit elle devenait vraiment excellente, soit les attaquants étaient de plus en plus discrets. Heureusement pour cette entreprise, ils ont su faire preuve d'humilité et refondre leur sécurité en partant du principe que leur système de détection était défaillant et qu'ils devaient le repenser plus finement, dans la réactivité et l'apprentissage des erreurs passées.



Face à des attaquants qui peuvent facilement avoir quelques longueurs d'avance sur le défenseur, il pourrait être intéressant de repenser la stratégie de sécurité, et la première brique, la définition de l'incident de sécurité, en partant de ce que l'on ne sait pas.

2 Vers une approche plus pragmatique : se poser les bonnes questions

2.1 Revenir aux fondamentaux opérationnels : « Dis-moi ce qui est important pour toi et je te dirai ce qui, pour toi, est un incident de sécurité »

Quel que soit l'état de la sécurité dans une organisation, on peut se livrer à un exercice tout simple : reprendre tout à zéro, et changer un peu de paradigme en répondant à quelques questions :

2.1.1 Quels sont les bijoux de ma couronne ?

Si un voleur peut tout me prendre, qu'est-ce qui me ferait le plus mal ? Si on a un peu de mal à aller à la pêche aux informations, une autre façon de poser la question sera : « Qu'est-ce qui me prendrait le plus de temps à reconstruire, à réobtenir, si demain tout était à refaire ? ». Une PME d'informatique vous répondrait peut-être « mon code source », un *hedge-fund* : « mon algorithme de *High-frequency trading* », ou une entreprise d'agro-alimentaire : « la confiance des clients ».

2.1.2 Qu'est-ce qui, dans mon activité, est le plus dépendant de mon SI ?

Quel est le chemin critique, servi par mon SI, entre ma mission et mon action ? Dit plus simplement, quelles sont les parties de mon SI que je dois maintenir en vie à tout prix si je veux continuer à accomplir ma mission ? Quel est le mode dégradé qui me permet quand même de maintenir mon activité ? C'est-à-dire, une fois dans ce mode dégradé, si quelque chose de supplémentaire tombe, je ne produis plus. Pour une banque d'investissement, ce sera son lien avec les places de cotation et le moyen de passer des ordres ; pour une chaîne hôtelière, ce serait son système de réservation et de revenue-management, etc.

À partir de ces deux premières questions, on peut en poser deux autres : à quel point suis-je aveugle ?

2.1.3 Quels sont les moyens dont je dispose pour savoir ce qui se passe dans mon SI ?

Suis-je capable de me rendre compte si mon ou mes *chemins critiques* sont compromis (attaqué, visé, reconnus) ou si on n'est pas en train d'essayer de me voler les bijoux de ma couronne ? C'est une question absolument critique, car elle suppose de se faire mal et de mettre en cause son système de fonctionnement. En effet, si le RSSI ne regarde rien, il ne détecte pas d'intrusion, et s'il ne détecte pas d'intrusion, le RSSI peut se dire (et laisser dire) qu'il fait un excellent travail. De plus, vu l'horizon de gestion de carrière des cadres, un RSSI change de poste, mettons, au bout de 3 ans alors que la fuite d'un élément de propriété intellectuelle peut se matérialiser au bout de dix ans. Si le RSSI est conscient d'un risque potentiel et qu'il fait son calcul, il arrive à l'alternative suivante : « *Mettre en place des moyens de détection plus pointus me coûte sur mon budget maintenant, alors que dans 2 ans et demi je suis parti et le problème surgira dans 5 ans au plus tôt... Pourquoi changer quoi que ce soit ?* ».

Note

Un esprit taquin signalait au rédacteur, lors de la relecture : « Il faut qu'il arbitre également de faire semblant de travailler ».

2.1.4 La question du « known good ». Connaître son système dans son état nominal

Si je suis en mesure de voir ce que se passe sur mon SI, suis-je en mesure d'appréhender si je suis en présence d'un événement adverse ? Inconnu ?

On aborde ici la question de la connaissance de son système dans son état de fonctionnement normal. On parle souvent de *baseline*. On évoquait dans la première partie une liste de « *known bad* » pour appréhender la situation dans laquelle on liste tous les événements adverses auxquels on peut penser. Maintenant si l'on veut se recentrer sur son propre système, si l'on veut détecter des événements dans son propre contexte, il faut commencer par investir dans la connaissance de son système en temps normal.

Attention, alors à ne pas aller trop vite en besogne, quand on évoque le fonctionnement « en temps normal » on ne doit pas forcément s'attacher au caractère sécurisé ou non de son système. En effet, connaître son système dans son fonctionnement nominal veut également dire : en connaître les vulnérabilités à un



moment donné. De là, idéalement on s'efforcera de déterminer les chemins d'attaques possibles.

Conclusion partielle : Avançons une définition opérationnelle et tirons-en les conséquences : « **Un incident de sécurité survient lorsqu'un événement de sécurité adverse et inconnu est détecté à l'endroit d'un item (bien) important pour le défenseur** ».

2.2 Que tirons-nous de cette définition ?

Reprenons la définition proposée juste au-dessus :

- « **Un item important pour le défenseur** ». Notre SI est cartographié, les applications répertoriées, et les bijoux de la couronne sont très clairement identifiés. Chacun a une image claire de son SI, à jour. Cela suppose aussi une connaissance de ses vulnérabilités et donc des chemins d'attaque envisagés.
- « **Est détecté** ». Le SI dispose des moyens de détection et de traçabilité tels qu'on est capable de stocker et conserver les traces nécessaires à une investigation ultérieure.
- « **Un événement inconnu** ». Là on peut considérer les deux termes. Tout d'abord, un événement. Ce serait une occurrence de sécurité. Mais un événement peut être connu. Dans le contexte d'un SI donné, le RSSI saurait ce qu'il a détecté, saurait exactement comment faire face, etc. Un incident de sécurité serait alors un item dont on ne comprend pas bien tous les contours. On saurait alors que c'est louche et que cela nécessite un effort supplémentaire de compréhension. En effet, un événement inconnu, aussi insignifiant qu'il puisse paraître, peut être le précurseur d'un incident de sécurité, critique.

Conclusion : On ne fera jamais l'économie d'un apprentissage de la « normalité » de son système. Mais l'étape suivante, pour être en ligne avec l'état de la menace, c'est d'être en mesure de détecter quelque chose qu'on n'est pas capable d'expliquer. C'est un changement de philosophie : aller à la chasse de ce qu'on ne comprend pas et, par itérations successives transvaser les éléments « inconnus » dans la catégorie « connus ».

3 Que faire ? Être constamment dans une démarche d'apprentissage et d'adaptation

L'incident de sécurité va en fait changer de nature en fonction de l'état de maturité de la chaîne de SSI et de réponse à incident d'une organisation. À l'image

d'une explosion, une organisation peut prendre en compte un incident à un moment différent selon son niveau de maturité et avoir affaire à différents niveaux de gravité, selon qu'elle est en mesure de prendre la mesure du problème plus en amont ; depuis la gestion des effets de la déflagration sans avoir rien su faire pour s'en prémunir, jusqu'à la suppression même de la possibilité d'une bombe.

En effet, on peut envisager les étapes suivantes qui correspondent à différents scénarios, chacun reflétant un niveau de maturité.

Des conséquences sans cause

La sécurité du SI est complètement immature, on ne voit rien, on ne sait rien, on se réveille un jour et on fait directement face à la matérialisation d'un événement adverse, sans en connaître ni les causes, ni le cheminement de l'attaquant, ni même la compréhension de l'existence d'un attaquant. Cela pourrait se matérialiser par la constatation par une entreprise qu'un concurrent autrefois complètement à la traîne grignote des parts de marché en raflant la mise sur des appels d'offres, sortant des produits ressemblant aux siens, etc. Il ne reste que les yeux pour pleurer. C'est la situation que l'on retrouve avec des organisations entièrement dépourvues de moyens de supervision ou de moyens dirigés contre des attaques ciblées. C'est également le cas auquel peut être confrontée une organisation qui est informée par une source extérieure d'une compromission (Agence étatique type ANSSI, autre organisation ou entreprise, etc.) et qui ne dispose d'aucun moyen technique pour exploiter ce signalement au sein de son propre SI.

Des causes sans incident

C'est le cas le plus couramment rencontré par les organisations dont la sécurité du SI n'est pas mûre, mais qui disposent de quelques moyens de supervision du réseau, de type NOC. Un exemple peut-être une sauvegarde qui se passe mal. Un analyste NOC y regarde de plus près, voit qu'un serveur de backup oublié au fin fond de la pampa est saturé. Il regarde encore d'un peu plus près et découvre un fichier .rar de 2 Go. Le fichier contient bien évidemment tout un tas d'informations stratégiques pour l'entreprise. « Ah tiens comme c'est curieux », au cours des deux années précédentes l'entreprise en question ne faisait que perdre les appels d'offres les plus critiques, toujours face au même genre de client. C'est la détection « par hasard », le niveau d'introspection est à peine meilleur que dans le cas vu plus haut, mais au moins on peut commencer à se poser des questions.

Des incidents inconnus

C'est la situation la plus courante. On se retrouve avec des indicateurs de compromission, mais on ne comprend pas exactement ce à quoi on a affaire. C'est un incident de sécurité, car on a détecté un événement adverse, mais avec suffisamment peu de données pour nécessiter des investigations complémentaires. Cela suppose d'avoir des moyens de supervision de la sécurité qui permettent



d'obtenir des bribes d'information : des traces de communications vers une IP curieuse, des événements de sécurité Windows étranges, une alerte d'un anti-virus, etc. C'est en fait toute situation face à laquelle l'équipe en charge de la sécurité du SI manque d'information pour gérer seule la situation. L'important à ce moment est, pour l'équipe de sécurité, d'avoir le réflexe et les moyens de se retourner vers une ressource à même de qualifier l'incident, d'en expliciter les contours de façon précise.

Des événements

Ici, on bascule dans la partie des entités qui commencent à prendre les devants. Un événement de sécurité recouvre l'ensemble des incidents connus. Dans ce cas, le problème existe, mais, contrairement à l'incident de sécurité, il n'y a pas de mystère pour comprendre ce qui se passe. Typiquement, un événement de sécurité serait ce qui arrive si une organisation doit, pour des raisons métiers, autoriser l'accès à un site web, mais que ce dernier se fait régulièrement compromettre. Elle met dans son IDS une signature levant une alerte en cas de tentative d'intrusion en provenance de ce domaine. Ce n'est pas un incident de sécurité, car l'équipe de sécurité sait ce à quoi elle fait face, mais une alerte surviendra de temps en temps, car elle ne peut pas couper définitivement l'accès à ce site web.

Des préventions

Là, c'est le Graal. L'organisation met en place des mesures pour que l'événement même n'apparaisse plus ou que ses conséquences soient immédiatement neutralisées. Ce pourra être le cas par exemple si l'organisation se met à travailler avec un système de *white-list* en messagerie ou navigation Web. Une autre version serait que, sur détection d'une alerte de l'anti-virus ou du HIPS, une station de travail soit automatiquement réinitialisée à partir d'une installation propre.

Passer de ne rien voir – si ce n'est les conséquences concrètes d'une intrusion – à tout détecter, et même avoir une politique de sécurité prévenant jusqu'à l'occurrence d'événements adverses. Ce processus peut être mis en place en détectant et remontant l'information inconnue, inexpliquée ; en menant une investigation, puis en reversant le produit de l'enquête pour que l'événement initialement inconnu rentre dans les pratiques de supervision et plus tard dans les politiques de sécurité pour que, *in fine* ce ne soit même plus un événement.

3.1 Des mesures organisationnelles pour un cycle efficace

On l'aura vu, passer de l'incident à l'événement requiert de recueillir, faire circuler et traiter de l'information. Du point de vue organisation, cela passe par identifier les acteurs de la SSI et savoir comment ils sont activés.

3.1.1 Qui prend les décisions et à quel stade de la crise et de l'investigation ?

Qui donne le « top » sur une action sur un SI ? Éteindre un serveur, couper un lien Internet peut avoir des conséquences irréversibles dans une crise et sur une investigation. On devra éviter de rentrer dans une crise sans connaître les règles d'engagement ni en ignorant « qui décide ».

3.1.2 Avoir des chaînes de communication internes rodées

Cette dernière doit être en mesure de recevoir et disséminer l'information importante en interne comme en externe (lien avec l'IT, la SSI, les autorités et les équipes expertes).

3.1.3 Avoir une logique d'escalade en place

À chaque moment, toute personne prise dans le cycle opérationnel de gestion d'une crise doit savoir vers qui se retourner en cas de doute, et qui est en attente d'une réponse de sa part. Pour aider à faire de cette idée une réalité, chaque maillon doit avoir conscience de l'existence des autres, des limites de leurs capacités et attributions. Cela suppose également le respect de chaque maillon et que chacun résiste à la tentation de court-circuiter des éléments du dispositif par volonté de « gagner du temps ».

3.2 Des mesures techniques pour détecter, analyser et réagir face à l'inconnu

3.2.1 Les outils d'implémentation de la politique de sécurité

Les concepts classiques ne sont pas à jeter par la fenêtre. Même si on a fait l'apologie dans la deuxième partie d'une meilleure connaissance de son propre système, une pierre angulaire reste encore de détecter et bloquer l'événement adverse. Car ne nous leurrions pas, comprendre c'est bien, empêcher c'est mieux.

3.2.2 Les outils de communication et de suivi

Même de rien, les outils de ticketing deviennent obligatoires. Difficile de survivre à un flot d'information et à ne pas « laisser de côté » une information critique



si l'on doit s'en tenir à son client de messagerie. Reste à trouver le système qui correspond à son organisation et ses moyens. L'écueil est bien évidemment la sur-ingénierie : monter un système hyper-complexe capable d'énormes prouesses, mais tellement lourd que les utilisateurs multiplieront les stratégies de contournement et l'outil deviendra un problème plus qu'une solution.

3.2.3 Les outils d'analyse et d'investigation

D'une part, les outils d'analyse doivent permettre, au plus près des outils de captation de l'information, de tamiser, de récupérer l'information utile à la conduite de l'investigation. Une bonne collaboration doit exister entre les équipes d'IT, de SSI et d'investigation pour optimiser les outils d'analyse. D'autre part, les outils d'investigation : plutôt que des outils complexes, la pratique tend à privilégier les outils spécifiques, à même de répondre efficacement à des questions précises.

Les outils

Seront – éventuellement – créés par les équipes d'investigation elles-mêmes en suivant le principe du « Simple & Secure ».

Charge à l'investigateur d'exploiter avec intelligence les résultats obtenus. On gardera dans cette catégorie le souci évoqué un peu plus tôt de développer les moyens d'exploiter des marqueurs qu'on aura pu obtenir auprès de sources externes telles que les agences gouvernementales ou d'autres organisations.

Les outils « OODA ». La boucle OODA est un concept apparu dans l'armée de l'air des États-Unis dans les années 1960, et qui caractérise le processus par lequel passe un pilote de chasse pour agir dans le cadre d'un combat aérien en itérant continuellement quatre processus : « *Observe, Orient, Decide and Act* » (« observer, s'orienter, décider et agir »). L'idée sera encore une fois de créer des outils au profit des équipes de sécurité au plus près de l'IT pour augmenter la capacité locale de réaction. Cela pourra être un script python fourni à des équipes de premier niveau de SOC et qui leur permettra d'aller rapidement interroger une base de données pour savoir si un domaine est malveillant ou pas et, si tel est le cas, générer automatiquement une règle Snort permettant de mettre à jour une sonde.

3.3 Exploiter des équipes expertes, dans une logique d'escalade

Dans une organisation où l'on se donnera les moyens de faire face à l'inconnu, et donc à l'incident de sécurité, on aura plusieurs étages :

- l'IT sera en mesure de mettre en œuvre ses outils et connaîtra bien son système. Cela permettra d'éviter de se retrouver dans la situation où on ne fait que se prendre l'événement adverse sans pouvoir comprendre d'où vient le coup. Ainsi, l'IT et la SSI connaîtront son système et son mode de fonctionnement nominal. Sans cette étape, toute protection est parfaitement illusoire ;
- la SSI sera en mesure de traiter les événements de sécurité et de remonter les incidents. Il faut être capable de relever les incidents de sécurité. La SSI doit maximiser ses moyens de détection et de stockage de traces (nous allons lire copieusement sur ce sujet dans les autres articles) ;
- un CSIRT (ou CERT selon les terminologies préférées) permettra de mener et coordonner les investigations pour comprendre un incident. Par la suite, les enseignements tirés de l'investigation iront en deux temps :

1. Faire de l'événement adverse inconnu, un événement adverse connu, c'est-à-dire un événement de sécurité que les équipes locales de SSI seront en mesure de traiter directement (avec des outils OODA, par exemple).
2. Dans un second temps, plus long, proposer des modifications de la politique de sécurité pour que soit mis en place les moyens techniques et opérationnels de sorte que l'événement de sécurité ne survienne même plus.

Conclusion partielle : Répartir des spécificités de sa propre organisation permet d'adopter une définition de « *l'incident de sécurité* » qui soit adaptée à soi. Cette approche permet de suivre son propre rythme, en fonction de ses moyens.

Conclusion, c'est l'incident de sécurité qui nous fait progresser

Ainsi, reste le plus important, l'humilité : accepter le fait que l'attaquant a la haute main et qu'il lui sera toujours plus aisé de faire son travail qu'à nous, défenseurs et investigateurs, de faire le nôtre. Comme le disait un sage Tibétain qui se reconnaîtra : « *le défenseur doit réussir 100% du temps pour gagner, l'attaquant ne doit réussir qu'une fois pour gagner...* ». Il est donc vital de bien communiquer sur la compréhension des attaques, tant dans les cercles techniques que managériaux ou décisionnaires, et de partager tant des outils que des méthodes. C'est la raison d'être de recueils tels que celui que vous avez en main. ■

ASPECTS JURIDIQUES DE LA RÉPONSE À INCIDENT

Florence Nnanga-Dupré – fld.dupre@gmail.com



mots-clés : DROIT / JURIDIQUE / PREUVE ÉLECTRONIQUE / DONNÉES PERSONNELLES / INCRIMINATION / LOGS DE CONNEXION / RESPONSABILITÉS

Qu'elle soit faite dans le cadre d'une prestation de service ou dans un contexte interne à une organisation, les équipes de réponse à incident sont souvent confrontées à différentes problématiques au rang desquelles figure l'épineuse question de la conformité légale et réglementaire des actions entreprises. En effet, quelle gouvernance faut-il mettre en place en amont ? Quels sont les standards et bonnes pratiques à appliquer ? Il s'agit, par ailleurs, pour ces équipes d'être en mesure de fournir des preuves admissibles en justice dans l'éventualité de poursuites judiciaires. Quelles sont les conditions pour une telle admissibilité dans un contexte, par définition immatériel et quel cadre juridique s'applique plus généralement aux activités de réponse à incident ? Autant de questions que nous allons tenter de démystifier ici, à la lumière des textes et référentiels juridiques applicables.

1 Introduction

De prime abord, il importe de définir ce que l'on entend par « incident » de sécurité. Cet exercice difficile est traité dans un article à lui seul. Pour les besoins de la cause, nous avons fait le choix de retenir la définition donnée par l'ISO 27000 en ces termes : « un ou plusieurs événements liés à la sécurité de l'information, indésirables ou inattendus présentant une probabilité forte de compromettre les activités de l'organisation et de menacer la sécurité de l'information ».

Dans ce contexte, certaines activités sont susceptibles de tomber sous le coup d'incriminations ou d'infractions diverses telles que l'usurpation d'identité, l'intrusion ou la tentative d'intrusion dans un système de traitement automatisé de données, le non-respect de règles et recommandations de sécurité et même aboutir à une mise en cause de la responsabilité de l'organisation que celle-ci soit contractuelle ou pas.

Il convient ici de souligner que cet article n'a pas la prétention d'être exhaustif et que seules certaines problématiques seront ici traitées.

De plus, et même si nous avons prévu de faire quelques références au droit international, notre référentiel sera plus principalement axé sur le cadre juridique européen et français.

2 Mesures préalables à la réponse à incident

Différentes actions doivent être entreprises et implémentées avant toute survenue d'un incident et partant, de réponse à incident, non seulement en termes de gouvernance, mais aussi d'un point de vue opérationnel, dans un but de maîtrise des risques en général et des risques juridiques en particulier.



2.1 La gouvernance

Côté gouvernance, au niveau étatique, européen et international, les organismes « certifiés » que sont les CERT communément appelés *Computer (Security) Incident Response Team*, constituent les premiers points de contact pour l'alerte et le traitement des incidents. Ils sont présents dans différents pays et majoritairement regroupés au sein du réseau FIRST (*Forum of Incident Response and Security Teams*). L'on peut ainsi citer le CERT-US, CERT-EU qui est l'organisme européen, le CERT-FR qui est l'organisme français, le CERT-UK etc., la plupart des états ayant leur organisme à cet effet, comme le requiert la réglementation européenne, en particulier la Directive du 7 février 2013.

Toujours en termes de gouvernance, la norme ISO 27035 préconise la mise en place d'une politique de gestion des incidents de sécurité qui s'intègre dans l'ensemble global des documents relatifs à la sécurité au sein d'une organisation. Cette politique doit contenir un certain nombre d'éléments parmi lesquels : une définition des incidents couverts, le cadre organisationnel et les responsabilités des différents intervenants.

Il est par ailleurs utile de rédiger une charte par laquelle les utilisateurs du SI, qu'ils soient employés, partenaires ou clients de l'organisation s'engagent à un certain nombre d'obligations dont celle de signaler toute faille de sécurité observée ou soupçonnée dans les systèmes ou services [1]. À ce sujet, il convient que l'information relative à l'organisation à contacter soit facilement accessible.

Par ailleurs, l'organisation doit mettre en place une équipe qualifiée, organisée, composée de compétences très spécifiques afin de permettre une réponse rapide et efficace. À ce propos, l'Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI) a publié, le 7 juillet 2014 un Référentiel d'exigences applicables aux Prestataires de Réponse à Incident de Sécurité (PRIS) en vue de leur qualification, dans lequel l'Agence définit les différentes fonctions d'une telle équipe et leur rôle au cours de la réalisation de la prestation de réponse à incident [2].

2.2 La nécessité d'un cadre contractuel adapté

Lorsqu'elle est réalisée par un prestataire externe, la réponse à incident doit être encadrée par le biais d'une convention signée entre le commanditaire et le prestataire [2]. En dehors de tout objectif de qualification, le référentiel précité, qui fait actuellement l'objet d'un appel à commentaires, peut être utilisé comme un guide de bonnes pratiques par les équipes de réponse à incident.

La convention signée entre le commanditaire et le prestataire prévoit un cadre strict des relations entre les parties. Celle-ci doit prévoir différentes choses, au rang desquelles figurent la confidentialité des informations transmises par le commanditaire et la répartition des responsabilités entre les parties. À noter que la Convention doit être signée préalablement à toute intervention du prestataire. En tout état de cause, il convient d'obtenir une autorisation du commanditaire ou du propriétaire du système. Et, dans le cas où la prestation débiterait avant la signature de la convention, le Référentiel prévoit, par exemple, que la confidentialité des informations transmises par le commanditaire doit être garantie au moyen d'un accord de confidentialité signé entre les parties.

2.3 Le rôle de la fonction juridique

D'une manière générale, certaines fonctions doivent être informées très rapidement après la survenue d'un incident de sécurité. Il s'agit de différentes fonctions qui interagissent et/ou interviennent en support de l'équipe de réponse à incident, au rang desquelles figure la fonction juridique. En effet, au même titre qu'une évaluation des risques techniques est nécessaire, le service juridique doit être impliqué en amont de la réponse à incident afin d'être en mesure d'évaluer les risques juridiques encourus. Ces deux analyses de risques combinées permettront, par ailleurs, d'entamer les discussions qui s'imposent côté assurance des risques cyber à mettre en place.

Enfin, outre la rédaction de la convention, le conseiller juridique pourra intervenir en support des équipes, tout au long de la prestation, sur les aspects de conformité légale et réglementaire des actions entreprises, la rédaction du rapport final ainsi que sur les suites judiciaires éventuelles. Il sera également, le cas échéant, l'un des interlocuteurs privilégiés des autorités et des instances judiciaires.

3 Les questions juridiques liées à l'accès aux systèmes utilisés par l'attaquant

Dans le cadre de la gestion d'un incident de sécurité, il peut s'avérer que les actes accomplis par l'attaquant d'une part, et ceux accomplis par les équipes d'intervention puissent tomber sous le coup de différentes incriminations. L'équipe de réponse à incident pourrait, par exemple, en suivant les traces



laissées par l'attaquant, remonter jusqu'aux serveurs de ce dernier et/ou dans les systèmes utilisés par l'attaquant à l'insu de leurs propriétaires légitimes. De tels actes pourraient alors tomber sous le coup de l'infraction d'accès et maintien frauduleux dans un système de traitement automatisé de données réprimé pénalement par différentes réglementations nationales.

Au niveau européen, le premier texte important que l'on peut citer dès lors que l'on parle de cybercriminalité, c'est la Convention de Budapest du 23 novembre 2001. La Directive relative aux attaques visant les systèmes d'information en date du 12 août 2013 [3] pourrait également trouver application dans ce contexte. En effet, il résulte de ce texte que les États doivent prévoir des sanctions dissuasives et proportionnées en ce qui concerne les attaques contre les systèmes d'information.

En France, les articles 323-1 et suivants du Code pénal issus de la loi Godfrain du 5 janvier 1988 [4] sont applicables. Ce texte réprime en réalité deux choses : l'accès frauduleux d'une part, et le maintien frauduleux d'autre part. Et, bien entendu, comme pour tout délit pénal, il faut que l'élément matériel et l'élément moral existent.

Par ailleurs, l'article 323-6 du même code prévoit la responsabilité de la personne morale, les sanctions pouvant aller jusqu'à l'interdiction de l'activité à l'occasion/dans l'exercice de laquelle l'infraction a été commise.

L'article 323-7 prévoit que la tentative des infractions prévues aux articles 323-1 à 323-3-1 est punissable. Ce dernier texte tel que modifié et inséré par la loi de Programmation militaire 2014-2019 dispose : « *le fait, sans motif légitime, notamment de recherche ou de sécurité informatique, d'importer, de détenir, d'offrir, de céder ou de mettre à disposition, un équipement, un instrument, un programme informatique ou toute donnée conçus ou spécialement adaptés pour commettre une ou plusieurs des infractions prévues par les articles 323-1 à 323-3 est puni des peines prévues respectivement pour l'infraction elle-même ou pour l'infraction la plus sévèrement réprimée* ». Cette nouvelle formulation a le mérite de spécifier clairement que la recherche ou la sécurité informatique constituent des motifs légitimes, et partant, sont des faits justificatifs.

4 Les problématiques relatives aux logs de connexion

Les données/ logs de connexion collectés, conservés et analysés constituent une source d'information importante pour les équipes de réponse à incident.

Mais l'utilisation et l'exploitation de telles données est soumise à de nombreux textes législatifs tant au niveau européen que national. Nous n'aborderons que quelques-uns de ces textes dans cet article.

Tout d'abord, la Directive du 15 mars 2006 sur la conservation des données de connexion [5] impose la conservation de certaines données de connexion, dans un but d'identification et à des fins d'enquêtes judiciaires. Il s'agit précisément des données permettant de pouvoir tracer et identifier la source d'une communication et sa destination, les données permettant d'identifier la date, la durée et l'heure d'une communication, le type de communication, la machine utilisée pour la communication, la localisation des équipements de communication mobile.

Cette directive exige une durée de conservation comprise entre 6 mois et 24 mois. Elle a été transposée en droit français par la Loi du 23 janvier 2006 [6] qui retient une durée de conservation d'un an.

La Cour de Justice de l'Union Européenne vient cependant de déclarer la directive précitée invalide eu égard notamment à la durée de conservation qu'elle juge « disproportionnée » [7]. La Cour relève cependant que la conservation des données en vue de leur transmission aux autorités nationales compétentes répond à un objectif d'intérêt général de lutte contre



NSC

No Such Con

2014

"What you're about to watch is a nightmare..."

WHERE & WHEN
Espace Oscar Niemeyer
19-20-21 November 2014

CONTENT
Top notch keynoters & speakers

OPEN BAR
Free beers, coffee and soft drinks for 3 days
*Save your money from cheap beers
to buy your ticket online!*

SOCIAL EVENT
Free beverages (beers & soft drinks), free food and party @La Rotonde [Place Stalingrad]

The bullshit-free conference www.nosuchcon.org



la criminalité grave et, partant, la sécurité publique. Mais elle excède les limites imposées par le principe de proportionnalité.

On peut alors s'interroger sur les implications d'une telle décision, en particulier au regard des réglementations nationales qui ont été mises en place pour transposer la Directive dans le droit national.

Par ailleurs, l'adresse IP fait l'objet d'un débat entre les experts, certains considérant qu'elle ne peut en aucun cas constituer une donnée personnelle, et par conséquent permettre d'identifier une personne, en l'occurrence un attaquant. D'autres, emboîtant le pas de la Commission Informatique et Libertés (CNIL) considèrent au contraire que ces données de connexion sont des données personnelles. Ainsi, par une ordonnance de référé du 17 juillet 2014, le Tribunal de Grande Instance de Paris a ordonné à la Banque Crédit Lyonnais de communiquer à une cliente qui le demandait les logs de connexion à ses comptes. Ce faisant, le Tribunal a considéré que les logs de connexion et l'adresse IP sont des données personnelles [8]. Ce statut de l'adresse IP comme donnée personnelle avait déjà été confirmé par l'Arrêt Promusicae de la Cour de Justice de l'Union européenne du 29 janvier 2008 [9].

5

Des obligations relatives aux données à caractère personnel et à la protection de la vie privée des personnes

Au cours des opérations de réponse à incident et des investigations, les équipes d'intervention pourront avoir accès à certaines informations relatives à la vie privée des personnes et/ou à des données personnelles y compris des correspondances. L'attaquant peut également avoir accédé à de telles données.

Dans ce cas, la réglementation sur la protection des données personnelles aura vocation à s'appliquer à côté des obligations relatives à la conservation des données de connexion. Il s'agira en fait de mettre en balance les obligations liées aux impératifs de sécurité et les obligations liées à la vie privée des personnes.

Au niveau européen, le cadre juridique pour la protection des données personnelles est basé sur la Convention 108 du Conseil de l'Europe de 1981, l'article 8 de la Charte des droits fondamentaux et les Directives du 24 octobre 1995, de 2002 dite e-privacy, de 2009 et le Règlement du 25 janvier 2012 adopté le

13 mars 2014. Les deux derniers textes introduisent entre autres des obligations de notification en cas de violation relatives aux données personnelles ainsi que le droit à l'oubli. En France, la loi Informatique et Libertés du 6 janvier 1978 modifiée prévoit un certain nombre d'obligations qui viennent s'ajouter aux obligations déclaratives. Tout d'abord, l'article 34 de ce texte prévoit une obligation générale de sécurité qui pèse sur l'entreprise dans la mesure où celle-ci doit garantir la sécurité des informations qu'elle recueille. Il ressort de ce texte que « *Le responsable du traitement est tenu de prendre toutes précautions utiles, au regard de la nature des données et des risques présentés par le traitement, pour préserver la sécurité des données et, notamment empêcher qu'elles soient déformées, endommagées ou que des tiers non autorisés y aient accès* ». La sanction prévue par l'article 226-17 du Code pénal en cas de manquement à cette obligation est de 5 ans d'emprisonnement et de 300 000 euros d'amende. Ce dernier montant est porté à 1 500 000 euros pour une personne morale.

L'enjeu ici est celui de la confiance et de la protection du patrimoine informationnel de l'entreprise, d'une manière générale.

Les équipes d'intervention pourront, par ailleurs, s'apercevoir que de telles données ont été détournées, ou font l'objet d'un vol ou de toute autre violation.

Dans ce cas, il existe des obligations de notification des violations aux données personnelles prévues par différents textes juridiques non seulement en France, mais également au niveau européen et que nous développerons dans la suite de notre exposé.

Enfin, les équipes de gestion des incidents de sécurité pourront, au cours de leurs opérations avoir accès aux correspondances/emails de certains utilisateurs du SI que ceux-ci soient des employés ou des partenaires de l'entreprise. La Loi du 10 juillet 1991 relative au secret des correspondances émises par la voie des communications électroniques pourrait, alors, avoir vocation à s'appliquer. Il résulte notamment de l'article 226-15 Code pénal modifié par la loi du 18 décembre 2013 que : « *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* ». L'alinéa suivant punit des mêmes 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 électronique ou de procéder à l'installation d'appareils de nature à permettre la réalisation de telles interceptions* » [10]. En outre, l'article 432-9 du Code pénal punit les mêmes faits, commis hors les cas prévus par la loi, par une personne dépositaire de l'autorité publique de 3 ans d'emprisonnement et de 45 000 euros d'amende.



Mais qu'est-ce qui relève du secret des correspondances et de la protection de la vie privée ?

Le principe c'est que la protection concerne tous types de courriers, électroniques ou pas (les e-mails, etc.). La jurisprudence est abondante dans ce domaine, le principe étant que tout ce qui est identifié comme « personnel » bénéficie de la protection. Et a contrario, ce qui a le caractère professionnel peut être consulté par l'employeur. En ce qui concerne le matériel utilisé, les fichiers ou courriers créés en utilisant le matériel informatique mis à disposition par l'employeur sont considérés comme la propriété de l'employeur [11].

6 Des obligations d'information et de notifications

Au niveau européen la Directive Sécurité des Réseaux et de l'Information (Directive SRI) adoptée récemment prévoit que les acteurs du marché doivent notifier à l'autorité compétente, sans retard injustifié, les « incidents qui ont un impact significatif ». Les incidents visés sont ceux qui portent atteinte à la sécurité et à la continuité d'un réseau ou d'un système d'information et qui entraînent une perturbation notable de fonctions économiques ou sociétales essentielles.

En France, l'article 34 bis de la Loi informatique et libertés introduit par l'Ordonnance n°2011-1012 du 24 août 2011 prévoit qu'en cas de violation de données à caractère personnel, le fournisseur de communication électronique accessible au public avertit, sans délai, la Commission Nationale de l'Informatique et des Libertés (la CNIL). Le texte prévoit également une notification à la victime, lorsque la violation porte atteinte à des données à caractère personnel ou à la vie privée d'une personne physique, sauf si le responsable du traitement parvient à prouver que des moyens appropriés ont été mis en place pour limiter l'impact négatif sur la vie privée.

Par ailleurs, l'article 22 de la Loi de Programmation Militaire 2014-2019 (LPM) du 18 décembre 2013 [12] met à la charge des Opérateurs d'Importance Vitale (OIV) l'obligation de déclarer, sans délai, au Premier ministre les incidents affectant le fonctionnement ou la sécurité de leur SI. Ce texte ne définit cependant pas la notion d'incident. Tout manquement à la loi est puni de 150 000 euros à 750 000 euros d'amende pour une personne morale.

Bien avant l'Europe, les États-Unis avaient déjà une réglementation incluant les « *data breach notification* ». Au pays de l'Oncle Sam, ces obligations de notification sont l'objet de nombreuses actions en justice et de demandes de réparations pécuniaires, sans compter les retombées négatives en termes d'image et de réputation.

7 Et la preuve dans tout ça ?

Dans le cadre de la réponse à incident, la collecte, l'acquisition, la conservation et la préservation de certaines informations récoltées constituent une étape importante dès lors que ces données peuvent être utilisées dans le cadre d'une action en justice. Se pose alors la validité de telles informations et de leur admissibilité en tant que preuve dans une instance judiciaire. Quel que soit l'environnement concerné, la preuve doit répondre à un certain nombre de critères parmi lesquels figure le principe de loyauté de la preuve qui conduit à écarter toute preuve obtenue de manière déloyale ou illicite ou au moyen d'une action non conforme à la loi.

De fait, outre ces conditions généralement requises pour l'admissibilité d'une preuve dite traditionnelle, le contexte « immatériel » rend plus complexes l'acquisition, la conservation et l'archivage des informations récoltées. En effet, le droit de la preuve a été longtemps marqué par la prédominance de l'écrit sous format papier. Le constat général qui est fait c'est que l'écrit électronique est plus fragile que l'écrit sur support papier, même si, aujourd'hui, l'équivalence entre les deux formes d'écrit est désormais affirmée par différents textes juridiques.

La première référence que l'on peut citer en droit français c'est la loi du 13 mars 2000 [13] transposant la Directive européenne du 13 décembre 1999 relative à la signature électronique [14] et son Décret d'application du 30 mars 2001. Cette Directive sera abrogée à compter du 1er juillet 2016 par le Règlement e-IDAS [15] adopté le 23 juillet 2014 qui pose, notamment, le nouveau cadre juridique de la signature électronique au niveau européen.

En France, l'article 1316 du Code civil issu de la loi du 13 mars 2000 admet la force probante de l'écrit électronique. L'article 1316-3 du même code, pour sa part, pose le principe de l'équivalence entre les deux formes d'écrit. Ce principe étant énoncé, il n'en demeure pas moins que certaines conditions doivent être remplies pour que la preuve électronique soit valide. Et c'est précisément l'objet de l'article 1316-1 qui prévoit d'une part que l'émetteur soit dûment identifié, d'autre part, que l'intégrité soit garantie. La signature électronique est le moyen préconisé pour remplir ces conditions.

Les équipes participant à une réponse à incident doivent veiller à préserver toutes les informations et les traces permettant de remonter jusqu'à la source des activités frauduleuses, à des fins éventuelles de poursuites judiciaires. La durée normale de conservation des données est d'un an. Cependant, cette durée peut être portée à 3 années lorsque ces données sont transmises aux autorités compétentes en vertu des dispositions de l'article 6 de la Loi pour la Confiance dans l'Économie numérique (la LCEN), du 21 juin 2004.

Il s'agit en particulier du cas du contexte de la lutte contre le terrorisme.

La collecte et la préservation des logs de connexion permettra par exemple d'être en mesure de fournir des informations utiles en termes d'imputabilité et donc de responsabilité d'une action sur le réseau ou le SI que celle-ci soit le fait d'un utilisateur interne ou externe ou même d'un attaquant.

À ce propos, la Norme ISO 27035 met l'accent sur l'aspect conservation des preuves afin de s'assurer que les éléments récoltés puissent être opposables dans le cadre d'une procédure judiciaire.

Dans le même sens, la Norme ISO 27037 relative à l'identification, la collecte, l'acquisition et la préservation des preuves numériques fournit un certain nombre de recommandations en direction de certains professionnels parmi lesquels les experts chargés de répondre à un incident informatique.

Enfin, lorsque des copies numériques sont effectuées, la Norme NF Z 42013 constitue une référence importante.

8 Des responsabilités...

8.1 Des actions possibles...

Nous l'avons dit, les éléments de preuve récoltés peuvent servir de base à une action juridique et/ou permettre à une organisation victime de s'exonérer, ne serait-ce que partiellement, de sa responsabilité, sous certaines conditions.

La responsabilité contractuelle de l'organisation qui a subi l'incident pourra également être recherchée par ses partenaires et contractants en cas de déni de service par exemple ou de vol/altération de leurs données au cours de l'incident.

8.2 Droit applicable et juridiction ?

De fait, l'enquête menée par les experts et les analystes forensiques peut conduire à une localisation de l'attaquant hors du territoire national, voire à l'autre bout du globe. On aura par conséquent un élément d'extranéité important, qui peut augurer une réponse juridique inefficace, car difficile à appliquer, du fait du principe de la territorialité des lois découlant du principe de souveraineté.

La question de la coopération internationale se pose alors, dans ce contexte, avec une acuité particulière. En effet, comment poursuivre l'attaquant que les équipes d'intervention ont réussi à localiser grâce à l'adresse IP et/ou les autres traces laissées dans le système ?

Par quel biais et selon quelles procédures ? Est-ce la loi du pays de localisation des serveurs, est-ce la loi du pays de résidence de la victime ? Bref quelle est la juridiction compétente ? On le voit, on est très vite confronté, dans ce cadre, à la question de la loi applicable.

Dans ce domaine, la Convention du Conseil de l'Europe signée à Budapest le 23 novembre 2001 sur la cybercriminalité constitue le texte de référence, dans la mesure où il posait déjà les jalons de la coopération internationale en matière de cybercriminalité. Cette nécessaire coopération est également affirmée dans les textes européens adoptés plus récemment et même au niveau international.

9 Les questions d'assurance et de garanties

Les informations récoltées scrupuleusement au cours des opérations de réponse à incident peuvent servir à l'organisation dans l'évaluation des risques. D'une manière générale, l'organisation sera en mesure de décider quels sont les risques à transférer à une assurance et les autres.

Par ailleurs et dans l'optique des discussions avec l'assureur, les éléments récoltés pourront servir de base de discussion pour le chiffrage de la prime d'assurance cyber risques à mettre en place.

10 Communication interne et externe

Outre les obligations de notification aux autorités nationales, en fonction du cadre réglementaire en vigueur et du secteur d'activités dans lequel opère l'entreprise et la notification aux tiers impactés par l'incident, il peut s'avérer utile de prévoir la communication de ce type d'événement tant à l'intérieur qu'à l'extérieur de l'entreprise.

En effet, afin d'éviter certaines conséquences négatives, mieux vaut mettre en place une politique de communication, en collaboration avec le responsable communication de l'organisation, avant toute survenue d'un incident de SSI.

Conclusion

On le voit, appréhender le cadre juridique de la réponse à incident n'est pas chose aisée, dans la mesure où il existe une pléthore de textes et réglementations ayant vocation à s'appliquer. La difficulté majeure vient du fait que les textes sont éparés et qu'il n'existe, à ce

jour, pas de code ou de droit spécifique applicable à la réponse à incident de sécurité, à notre grand regret. Le critère principal pour identifier la règle de droit qui s'applique reste la caractérisation préalable des opérations envisagées et/ou le type de données concernées. Dans un tel contexte et afin de mener ce type d'opérations dans un cadre garantissant la plus grande sécurité juridique, il convient de s'entourer de praticiens du droit aux côtés des opérateurs techniques, et ce, durant tout le cycle des opérations. ■

■ RÉFÉRENCES

- [1] <http://www.clusif.asso.fr>, « Gestion des incidents de sécurité du système d'information (SSI) », Les dossiers techniques du CLUSIF (Club de la sécurité de l'information français), mai 2011
- [2] Référentiel d'exigences applicables aux prestataires de réponse à incident, version 0.3 du 7 juillet 2014 publié avec un appel à commentaires
- [3] Directive n° 2013/40/UE du Parlement et du Conseil du 12 août 2013 relative aux attaques contre les systèmes d'information et remplaçant la Décision-cadre 2005/222/JAI du Conseil du 24 février 2005
- [4] Loi n°88-19 du 5 janvier 1988 relative à la fraude informatique
- [5] <http://www.eur-lex.europa.eu>, Directive 2006/24/CE du Parlement Européen et du Conseil du 15 mars 2006 sur la conservation des données générées ou traitées dans le cadre de la fourniture de services de communications électroniques accessibles au public ou de réseaux publics de communication et modifiant la directive 2002/58/CE
- [6] Loi n°2006-64 du 23 janvier 2006 relative à la lutte contre le terrorisme et portant des dispositions diverses relatives à la sécurité et aux contrôles frontaliers, JO 24 janv., p1129
- [7] Arrêt CJUE n° C-293/12 du 8 avril 2014
- [8] TGI Paris référé, 17 juillet 2014
- [9] Article Legalis.net du 23 juillet 2014
- [10] Loi n°2013-1168 de Programmation Militaire 2014-2019, du 18 décembre 2013
- [11] Cass.soc, 19 juin 2013, n°12-12.138, Young et Rubicam c/M.P
- [12] Loi n°2013-1168 du 18 décembre 2013 dont les décrets d'application sont en cours de rédaction
- [13] Loi n°2000-230 du 13 mars 2000 portant adaptation du droit de la preuve aux technologies de l'information et relative à la signature électronique
- [14] Directive n° 99-93 du 13 décembre 1999
- [15] Règlement du Parlement Européen et du Conseil sur l'identification électronique et les services de confiance pour les transactions électroniques au sein du marché intérieur

NE MANQUEZ PAS LE DEUXIÈME NUMÉRO !

**DÉMONTEZ !
COMPRENEZ !
ADAPTEZ !
PARTAGEZ !**



**DISPONIBLE CHEZ
VOTRE MARCHAND
DE JOURNAUX ET SUR :
boutique.ed-diamond.com**

ANALYSE DE MALWARE À LA RESCOURS DU CSIRT : DE LA RÉTRO- CONCEPTION AUX IOC

Mouad Abouhali – mouad.abouhali@airbus.com (@_m00dy_)

Security Researcher at Airbus Group Innovations

mots-clés : MALWARE / RÉPONSE À INCIDENT / INDICATEURS DE COMPROMISSION / LABORATOIRE / ANALYSE STATIQUE / DYNAMIQUE / PROTECTIONS

Cet article a pour objectif de proposer une démarche d'analyse de malware dans le cadre d'une réponse à un incident. L'analyste doit adopter une approche rigoureuse afin de qualifier la menace inhérente à cet incident tout en agissant dans un temps limité. Son objectif est de produire une analyse exhaustive du malware comprenant les marqueurs nécessaires pour contenir et éradiquer l'incident remonté.

1 Introduction

Selon le **[NIST]**, le cycle de vie d'une réponse à un incident se compose de quatre étapes essentielles : la préparation, la détection et l'analyse, l'éradication et le retour à un état stable et enfin la capitalisation. La phase de détection et d'analyse est souvent menée par une structure de type CSIRT (*Computer Security Incident Response Team*). Cette dernière est amenée à faire appel aux compétences d'un spécialiste dans le cadre d'une analyse de malware.

Le présent article s'attachera à proposer une méthodologie qui peut être adoptée et adaptée par toute entité qui vise à intégrer l'analyse de malware dans son processus de gestion d'incident.

Cette analyse se focalise essentiellement sur l'étude d'un binaire supposé malveillant et élément clé de l'incident de sécurité. Elle a pour objectif d'établir les fonctionnalités du binaire (*downloader*, porte dérobée, RAT, Bot, *ransomware*, etc.) et d'identifier les indicateurs de compromissions (IOC : *Indicators Of Compromise*). En fonction du contexte de l'incident, le CSIRT peut spécifier sa demande en indiquant à l'analyste les éléments qu'il faudra vérifier en priorité.

Les IOC correspondent à l'ensemble des *artefacts* ou caractéristiques qui permettent d'identifier d'une manière unique et sûre un malware. Ces caractéristiques peuvent correspondre par exemple aux paramètres utilisés par le malware pour communiquer avec un serveur de contrôle externe (nom de domaine, adresse IP, etc.), ou aux éléments mis en place pour assurer sa persistance.

Ces IOC vont être utilisés pour générer les différentes signatures nécessaires pour l'éradication de la cause de l'incident et cela en alimentant les différentes sondes réseau et système, déjà présentes au niveau du SI ou nouvellement installées suites à l'intervention du CSIRT.

Nous allons considérer que le processus d'analyse proposé aura comme éléments en entrée : le(s) binaire(s) malveillant(s) et en sortie un rapport comprenant des IOC et une description précise des caractéristiques du malware. Le processus d'acquisition de ce(s) binaire(s) ainsi que le traitement des IOC ne seront pas abordés au niveau de cet article.

Considérant que les IOC vont constituer un outil de travail pour le CSIRT pour la suite du traitement d'incident, ils vont ainsi établir le fil conducteur de notre analyse. Ces derniers peuvent être formalisés sous le format d'un simple fichier texte, ou transcrits dans des frameworks de type **[openIOC]** ou **[YARA]** (réutilisables par les sondes).

Généralement, le processus d'analyse devra être piloté par un objectif qu'est l'exhaustivité de l'analyse tout en gérant les différentes **contraintes organisationnelles et techniques** :

- au niveau organisationnel, l'analyste doit agir en fonction du temps alloué, des ressources disponibles et de la liste des malwares en attente d'analyse ;
- au niveau technique, l'analyste devra faire face à des mécanismes de protection déployés par le malware pour assurer sa furtivité et retarder sa détection.

L'analyste devra donc jongler avec ces contraintes en veillant à alimenter, au fur et à mesure, l'équipe

de réponse à incidents des éléments (IOC) qui vont servir à identifier, contenir et éradiquer l'incident.

La bibliographie et les ressources Internet traitant l'analyse de malware sont très riches **[PRACTICAL]**, **[OPENSEC]** et **[BLOGFU]**. Elles incluent les détails techniques des notions évoquées tout au long de cet article que ce soit au niveau de l'architecture d'un système, l'utilisation d'un outil, ou la description d'une technique donnée. De ce fait, cet article vise à introduire les notions de base d'analyse de malwares aux lecteurs désirant découvrir le domaine. Le *Malware Corner* de MISC représente un excellent moyen d'étudier les différentes techniques employées par les malwares sous une vision différente à chaque fois.

L'outillage sera abordé d'une manière globale, et le choix des outils sera laissé au lecteur en fonction de ses affinités. Cela dit le site web de **[WOODMANN]** reste un très bon point d'entrée.

Enfin, la démarche proposée tout au long de cet article s'articulera essentiellement autour des binaires Windows, mais rien n'empêche de la réadapter à d'autres types de binaires et systèmes d'exploitation.

2 Méthodologie de travail

La méthodologie proposée se base essentiellement sur deux approches :

- une analyse de code non-exécuté (ACNE) (approche statique) : qui vise à inspecter la structure et le code du malware sans pour autant l'exécuter. Elle s'appuie essentiellement sur une lecture du code désassemblé voire du code source construit à partir du code machine ou du Byte code d'une machine virtuelle (JVM, .NET, etc.) ;
- une analyse comportementale (approche dynamique) : qui vise à étudier les interactions (mécanismes d'exfiltrations de données, persistance, etc.) du malware avec le monde extérieur, notamment le système d'exploitation et la couche réseau.

Contrairement à l'analyse statique de binaire, qui repose beaucoup sur des outils manuels, l'analyse comportementale, quant à elle peut être automatisée. Nous noterons néanmoins que les résultats fournis par les outils automatisés sont généralement insuffisants et nécessitent d'être complétés par une étude manuelle.

L'analyste aura besoin de constituer une boîte à outils dont le socle de base comporte des désassembleurs (IDA Pro, Hopper, etc.), des débogueurs (Windbg, Ollydbg, Immdbg, gdb, etc.) et des éditeurs de binaires (CFF explorer, HTE, etc.).

2.1 Laboratoire d'analyse

Notre laboratoire d'analyse sera constitué principalement de machines virtuelles, le choix technologique du produit (Vmware, Xen, Virtual PC, etc.) sera laissé au lecteur

selon son retour d'expérience. La virtualisation permettra une exécution du malware dans un environnement maîtrisé afin d'assurer un confinement du laboratoire.

La virtualisation offre un moyen pratique pour analyser un malware s'installant au niveau du noyau du système d'exploitation. Il est possible d'attacher facilement un débogueur noyau à la machine d'analyse depuis une seconde machine (virtuelle ou physique).

Dans certains cas très rares, il est recommandé d'avoir une machine physique dans son laboratoire dans le cas où le malware ne fonctionnerait pas correctement sous une machine virtuelle et que l'analyse ne permettrait pas d'identifier les mesures employées par le malware pour se protéger contre une exécution contrôlée.

Notre laboratoire peut être complété par une brique qui assurera le lien entre l'analyse de code non-exécuté et l'analyse dynamique. Cette brique peut être implémentée sous forme d'un outil qui vise à envoyer les événements de débogage depuis la machine d'analyse vers un désassembleur présent sur une autre machine. Le travail réalisé au niveau de la partie dynamique sera ainsi transcrit vers la partie statique (en coloriant par exemple les chemins d'exécutions, ou en ajoutant des commentaires). L'outil **[Qbsync]** constitue un excellent allié pour ce type de manipulation.

Une autre approche se base sur l'exploitation de la trace d'exécution. Cette approche vise à exécuter un malware dans un environnement cloisonné, puis de récupérer une trace de son exécution afin de l'analyser en se focalisant sur un chemin d'exécution **[pTra]**.

Des solutions « prêtes » à l'utilisation existent comme Cuckoo Sandbox, Joe SandBox, Anubis, FireEye, IRMA, etc. Ces produits peuvent constituer une solution optimale pour un traitement automatisé des malwares.

2.2 Modus operandi : processus d'analyse

En plus de son exhaustivité, le processus d'analyse doit être répondre aux besoins de l'équipe de réponse à incident qui peut demander une analyse complète ou des IOC précis (nom de domaine, moyen d'exfiltrations, technique de persistance, etc.).

Toutefois, quelques règles de bon sens peuvent être prises en compte :

- la connaissance de l'implémentation, à l'aide des API Windows, des différentes techniques utilisées par les malwares pour réaliser leurs actions classiques : persistance, capture des frappes clavier, récupération des mots de passe, injection de code, etc. permet généralement à l'analyste de localiser les sections de code intéressantes ;
- les indicateurs identifiés durant l'analyse comportementale peuvent être utilisés pour orienter le processus de lecture de code. Nous

considérerons par exemple l'accès en lecture à une clé de registre comme un point de départ ou d'entrée pour notre analyse du code ;

- comprendre le code dans sa globalité et ne pas perdre du temps à essayer d'interpréter chaque instruction. L'idéal est de « zoomer » sur les sections de code correspondant à des fonctionnalités sensibles. Cela rejoint globalement les techniques de rétro conception, par exemple nous n'allons pas nous attarder à disséquer un algorithme cryptographique standard.

2.2.1 Processus de capitalisation

Avant de « zoomer » sur les différentes recettes d'analyse de malware, il est intéressant de rappeler l'importance de maintenir un processus de capitalisation des éléments produits durant chaque analyse.

Selon le schéma suivant (Fig. 1), une analyse de malware doit nous permettre de construire les IOC correspondants à l'incident courant, mais en même temps ses résultats doivent être classés puis stockés par exemple au niveau d'une base de données.

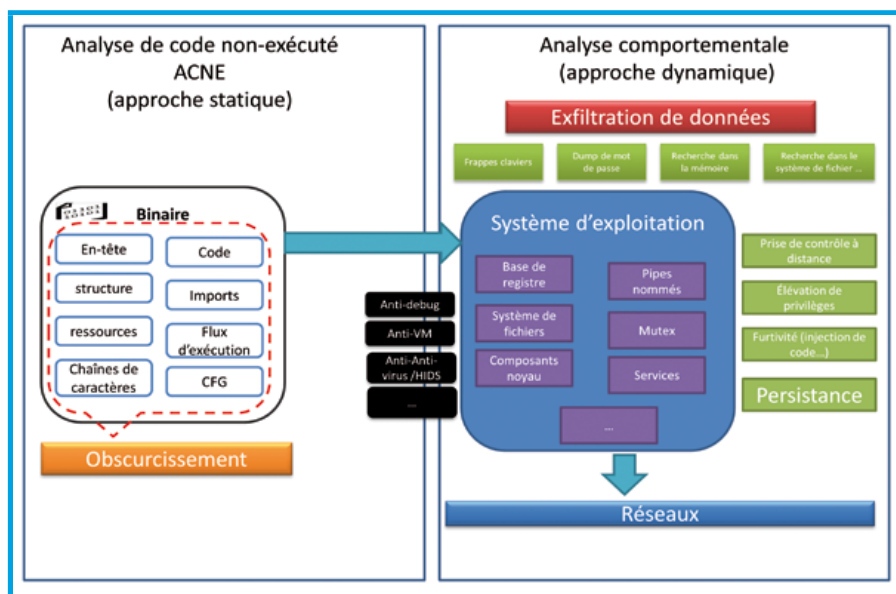


Fig. 1 : Le cycle d'analyse de malware.

2.2.2 Analyse dynamique vs analyse statique

Avant de foncer « tête baissée » dans une analyse, nous devons nous assurer que le malware à étudier ne présente pas des similitudes avec un binaire déjà présent dans notre base. L'analyste peut aborder la recherche des similitudes soit naïvement en comparant les caractéristiques du malware avec ceux de la base de données (ou même en effectuant un *binary diffing*), soit idéalement en s'appuyant sur des outils automatisés comme [VxClass], [YARA].

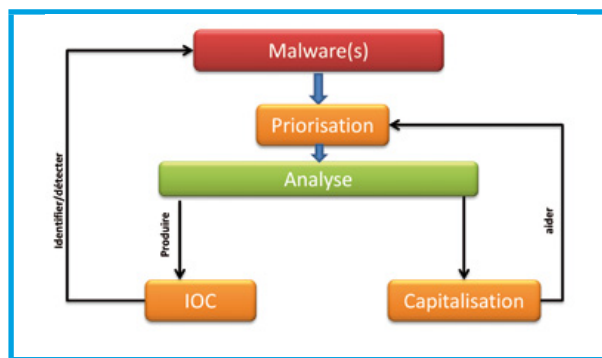


Fig. 2 : Vue macroscopique du processus d'analyse de malware.

Cette démarche peut se révéler très efficace dans le cas où le malware correspond à une variante d'un malware déjà analysé. Dans ce cas de figure, l'analyste n'aura qu'à déceler les différences (nouveau nom de domaine, nouvelle adresse IP, etc.), et pourra éventuellement réutiliser des outils développés lors d'une précédente analyse.

Une fois que les résultats de cette première étape nous confirment la nouveauté du malware, nous pouvons dérouler notre analyse, représentée par le graphique suivant en Figure 2.

Évidemment, les deux approches (statique et dynamique) vont se heurter à un moment ou un autre à des mécanismes de protections (*packing*, anti-débogage, anti-virtualisation, anti-désassemblage, etc.) et d'obscurcissement (encodage, chiffrement, etc.) qui vont rendre la tâche plus ou moins complexe à l'analyste. Et en fonction de la nature de l'attaque, la valeur de la cible, le niveau de furtivité visé, les auteurs de malwares ont tendance à empiler plusieurs couches de protections afin de rendre la tâche encore plus difficile.

L'analyse va se construire autour des différentes hypothèses émises au fur et à mesure par l'analyste. De ce fait, un aller-retour entre les deux approches se fera naturellement afin de confirmer ou infirmer ces hypothèses.

2.2.3 Un mot sur l'exhaustivité de l'analyse

Plusieurs indicateurs peuvent être mesurés pour assurer l'exhaustivité de l'analyse si nécessaire :

- la couverture du code : sa mesure peut être un bon indicateur de l'exhaustivité de l'analyse. Elle

peut être calculée en instrumentant le malware de manière à atteindre l'ensemble des fonctionnalités identifiées (chemins d'exécution). Une technique basée sur PIN est décrite ici **[Code Coverage]** ;

- identification de toutes les fonctionnalités du malware : si le malware se connecte à un C&C et reçoit des commandes, il serait intéressant de développer un C&C et envoyer ces mêmes commandes une à une et valider le comportement du malware.

3 Recherche initiale d'informations

3.1 Fiche d'identité

Cette étape va permettre à l'analyse de prendre en main le malware à étudier tout en construisant sa fiche d'identité qui englobe à la base les informations suivantes :

- l'empreinte MD5 ou SHA du binaire ;
- la date de compilation ;
- des informations sur la chaîne de compilation ;
- la structure du binaire : en-tête et sections ;
- l'entropie du binaire et de ses différentes sections.

À ce stade, nous pouvons envisager d'envoyer l'empreinte (hash) du malware à un site de reconnaissance de malware tel que VirusTotal afin de récupérer des éléments d'analyse effectués par ces entités. Toutefois, cette pratique doit être conforme à la politique de sécurité établie au niveau de l'entreprise cible de l'incident.

3.2 Un premier coup de microscope

3.2.1 Les en-têtes et sections du binaire

Des erreurs intentionnellement insérées par l'attaquant au niveau des en-têtes du PE peuvent induire en erreur les outils d'analyse. De ce fait, inspecter manuellement la structure du PE permet aussi de relever ce type d'anomalie et cela en s'appuyant sur des outils comme **[CFF explorer]**, **[xPELister]**.

3.2.2 La table des imports et API utilisés

Un malware s'appuie généralement sur les bibliothèques déjà présentes au niveau d'un système d'exploitation pour effectuer ses actions (communication réseau, chiffrement, etc.). Cependant, l'auteur du malware peut décider de ne pas exposer ce type d'information.

Il est commun de constater que les API nécessaires au malware sont chargées à l'exécution par le biais d'un appel aux fonctions **LoadLibrary** et **GetProcAddress** dont les paramètres de type chaînes de caractère (nom de la DLL et le nom de la fonction) peuvent être aussi obscurcis par un simple encodage ou du chiffrement. L'article **[API_OBUFSICATION]** regroupe les différentes techniques utilisées dans ce cadre.

À l'instar de **[ImpREC]** et **[ChimpREC]** qui visent à reconstruire la table des imports d'un binaire PE une fois que ce dernier est « *mappé* » au niveau de la mémoire, **[Scylla]** présente quelques nouveautés utiles.

3.2.3 Les chaînes de caractères

Les chaînes de caractères peuvent s'avérer très utiles tout au long d'une analyse. Leur présence peut d'une part fournir des indications sur les sections de code à privilégier et d'autre part, récupérer certains paramètres du malware (nom de domaine du C&C, nom de fichier, etc.). Leur absence peut également soulever plusieurs hypothèses et notamment l'utilisation d'une couche d'obscurcissement.

L'auteur du malware peut décider de déchiffrer les chaînes de caractères d'un seul coup, ou à la volée en évitant que ces dernières ne soient divulguées lors d'un dump de la mémoire.

Concernant l'obscurcissement complexe ou « *fait maison* » des chaînes de caractères, nous allons privilégier, au vu du facteur temps, des méthodes visant à réutiliser le code du malware au lieu d'analyser, comprendre puis réécrire l'algorithme dans un autre langage. Nous pouvons garder cet exercice pour la fin de l'analyse afin de produire un décodeur réutilisable dans d'autres cas.

Pour cela, nous pouvons nous appuyer sur les techniques d'instrumentation de binaire comme l'utilisation de la fonctionnalité **[AppCall]** d'IDA, qui permet d'invoquer une fonction présente dans le binaire avec les paramètres souhaités, notamment depuis un script Python. Le choix est laissé au lecteur parmi le panel des outils d'instrumentation de binaire (**[Pin Tools]**, **[Miasm]**, **[ImmDbg]**, etc.).

3.3 Identification des mécanismes de protection

3.3.1 Technique de *packing*

Il faut procéder à une vérification du binaire afin de valider si ce dernier est protégé. Plusieurs symptômes peuvent aiguiller l'analyste vers cette hypothèse. Par exemple : la proportion du code désassemblé par rapport à la taille du binaire, taille et nom d'une

section, la table d'import réduite, absence de chaînes de caractères, sans négliger bien sûr le comportement de certains désassembleurs et débogueurs face à ce type de binaire.

L'*unpacking* reste tout un art à lui tout seul, mais généralement l'analyste commencera par identifier la nature du *packer* utilisé (produit connu ou fait maison). Selon le type du *packer*, il peut opter pour une recherche sur Internet d'un outil ou une *recette* générique ou pour étude du code du *stub* en charge de l'*unpacking* afin de développer un script similaire au *stub* ([**Miasm**], [**Metasm**], etc.). Comme il peut tenter de *dumper* le binaire *mappé* en mémoire une fois ce dernier *unpacké dynamiquement* en utilisant par exemple : [**ChimpRec**] ou [**OllyDump**].

3.3.2 Protections dynamiques

Afin d'assurer que son exécution n'est pas passée à la loupe, un malware cherchera à vérifier si certains outils d'analyse ne sont pas installés ou exécutés. Cela peut être fait de différentes manières :

- inspecter les pipes nommés créés par les outils de [**Sysinternals**] ;
- parcourir la liste des processus à la recherche de processus suspects ;
- parcourir les objets Windows (Mutex, port de communication, drivers, etc.), en recherchant des chaînes de caractères au niveau de la mémoire ;
- invoquer des requêtes WMI (vers **Win32_Processor**, **Win32_SCSIController**, **Win32_ComputerSystem**, requêtes IOCTL, etc.).

Cependant et en plus de l'aspect purement système (processus, caractéristiques du matériel, type de la carte réseau, clefs de registres), le malware peut déployer aussi des techniques dites [**anti-virtualisation**] visant à modifier son comportement.

Les techniques anti-débogage visent quant à elles à ralentir l'analyste dans son processus de débogage. En la matière, le papier du chercheur [**Peter Ferrie**] reste une référence et reprend les techniques les plus connues depuis l'utilisation des API Windows, jusqu'aux techniques les plus élaborées.

Le contournement de ces techniques peut être effectué en s'appuyant sur des « plugins » proposés pour différents débogueurs (exemple : [**OllyAdvanced**] pour OllyDbg). Toutefois, l'idéal reste de s'imprégner de la technique utilisée afin de produire soi-même le moyen de contournement : développement d'un script qui intervient au moment du débogage ou qui modifie le code binaire du malware. Cela permettra à l'analyste de produire sa propre boîte à outils qu'il/elle peut faire évoluer en fonction des techniques nouvellement rencontrées.

4 Salle d'observation : analyse comportementale

Dans le cadre d'une réponse à un incident, il est primordial de fournir à l'équipe en charge de contenir l'attaque, des IOC dans un temps généralement réduit. Or, certains indicateurs peuvent être facilement récupérés par une analyse comportementale là où une analyse de code peut se révéler consommatrice de temps.

Il est recommandé de démarrer l'analyse sur la base d'une machine « propre » et dont les différents composants et outils installés sont connus et maîtrisés. Cela aidera l'analyste à repérer un comportement nouveau au socle de base de la machine de référence en éliminant facilement le « bruit » provenant des composants de base.

L'approche proposée à ce stade est de lancer le malware après avoir activé des outils de supervision des événements système qui vont se charger de capturer et d'enregistrer toutes les actions effectuées par le malware et qui ont un impact sur le système d'analyse. Pour cela, la suite [**Sysinternals**] de Microsoft et leur nouvel outil [**SysMon**] restent indispensables. L'outil [**Capture-BAT**] propose aussi cette fonctionnalité en générant un fichier log et une capture réseau des paquets potentiellement émis par le malware, sans passer par une GUI.

Pour ceux qui préfèrent analyser les événements collectés sous une vision graphique, l'outil [**ProcDot**] du CERT autrichien permet d'agréger l'ensemble des événements enregistrés par l'outil [**ProcMon**] et les captures des paquets réseaux pour produire un graphe.

Parfois, la simple exécution du malware n'implique aucune réaction notable. En effet, le malware peut vérifier quelques éléments de son environnement d'exécution, nécessite des paramètres passés en ligne de commandes, ou attend simplement une action humaine (exemple : bouger le pointeur de souris).

4.1 Émulation d'un environnement réel

Suivant la même logique de récupération rapide et efficace d'IOC et afin de faciliter la phase dynamique de l'analyse, il est préférable de connecter la machine infectée à une deuxième machine qui émule l'ensemble des services réseaux demandés par le malware. L'analyste peut opter pour des outils tels que [**Inetsim**], [**Honeyd**], [**Remnux**] ou produire ses propres scripts.

En fonction des résultats obtenus, paquets réseaux ou événements systèmes, il sera ensuite possible de localiser les appels systèmes afin de mieux comprendre l'origine de ces événements.

La construction de cet environnement va être aussi conditionnée par les résultats de notre analyse statique,

en fonction des besoins du malware (connexions à un service TCP spécifique, vérification d'un en-tête HTTP, valeur d'un paramètre donné d'une réponse HTTP, etc.).

Généralement, les données récupérées depuis un environnement réel peuvent se révéler d'une grande aide à l'analyste qui pourra ainsi les confronter avec ses conclusions.

Afin d'assurer sa persistance, le malware sauvegarde parfois ses paramètres de configuration au niveau de la base de registre. L'analyste peut se focaliser sur les événements impactant ce composant en réalisant par exemple un différentiel entre le contenu initial de la base de registre et son contenu après exécution du malware (en utilisant l'outil **[RegShot]** par exemple).

La supervision des appels aux API Windows peut se révéler très efficace pour comprendre le fonctionnement du malware et identifier les paramètres passés aux API. L'analyste peut avoir recours à un outil tel que **[ApiMonitor]**, ou passer par une instrumentation du malware. Là aussi le malware peut se protéger en vérifiant par exemple si la dll **dbgHELP.dll** est présente au niveau de sa mémoire, ou invoquer directement des fonctions du noyau s'il dispose des privilèges nécessaires.

Dans certains cas, le malware est exécuté par un autre binaire (**Launcher**) qui lui a transmis les paramètres nécessaires à son bon fonctionnement et qui correspondent aux coordonnées du C&C par exemple. Cela peut amener à une analyse incomplète si l'analyste n'accède pas à cette information. Généralement, l'analyste peut demander à ce stade un dump de la mémoire, ou le contenu d'une clé registre voire un fichier présent sur la machine initialement infectée afin de correctement enclencher certaines fonctions du malware.

5

Une bonne crise d'ACNE : analyse du Code Non-Exécuté

La complexité de l'analyse statique est fortement liée aux connaissances de l'analyste du domaine de la rétro conception et à la technologie de programmation utilisée pour l'élaboration du malware. La reconnaissance rapide de certains motifs de code permet de réduire le temps d'analyse en écartant les portions de code propres au langage de développement et en se concentrant sur l'analyse de la charge utile.

5.1

Bibliothèques et outils embarqués

Dans certains cas, le malware est lié statiquement à des bibliothèques (exemple : OpenSSL). Cela implique

que l'analyste va se trouver face à un binaire de taille et de complexité importantes. L'objectif ici est de réduire le temps d'analyse en isolant le code de la bibliothèque utilisée et de restreindre l'étude au code du malware.

Pour cela, l'une des techniques proposées nécessite en premier lieu d'identifier la bibliothèque utilisée et sa version (en inspectant les chaînes de caractères par exemple). La deuxième condition concerne l'accès au code source de cette bibliothèque et idéalement de connaître le compilateur qui a été utilisé pour produire le binaire final. Si toutes ces conditions sont réunies, il est alors possible de compiler soi-même cette bibliothèque et puis effectuer un différentiel entre le malware et le binaire de la bibliothèque compilée. L'outil le plus connu pour réaliser cette manipulation reste **[BinDiff]** de Dyanamics et son plugin IDA qui permet d'importer les symboles des fonctions présentant des similitudes. IDA propose aussi sa base de signature **[FLIRT]** qui contient les signatures des fonctions de différentes bibliothèques. Cette base de données peut être étendue en utilisant les outils **[FLAIR]**.

5.2 Les motifs de construction de code

La capacité de l'analyste à reconnaître rapidement certains motifs de code va lui permettre de mieux cartographier le code du malware. Ci-dessous une liste non exhaustive de certains motifs de code assez récurrents :

- algorithmes cryptographiques : leur identification peut s'appuyer sur la détection de la bibliothèque cryptographique utilisée (OpenSSL, CryptoAPI, etc.), la recherche de signatures et constantes propres à ces algorithmes en utilisant des outils tels que **[FindCrypt]**, **[IDA_Signsrch]** ou **[PeID Krypto Analyzer]**. Généralement, l'utilisation des algorithmes de hachage respecte la même logique : une fonction d'initialisation d'un contexte, la mise à jour de ce contexte et une fonction de finalisation **[SSTIC_CRYPT0]**.
- techniques d'injection de code : le malware pourra corrompre le système cible en injectant son code au niveau des processus systèmes (utilisation de **OpenProcess**, **CreateRemoteThread**, **WriteProcessMemory** et **QueueUserAPC**). En fonction des privilèges dont il dispose, il cherchera à modifier les structures du système (exemple : DKOM, modification de la SSDT, installation d'un pilote, accès au Device\PhysicalMemory, etc.) pour assurer sa furtivité ;
- techniques de recherche manuelle de DLL : utilisée généralement par les shellcodes, cette technique a pour objectif de localiser dans un premier temps **kernel32.dll**, puis parcourir la structure PE de cette dernière à la recherche des fonctions **LoadLibrary** et **GetProcAddress** (accès à la liste doublement chaînée des modules depuis le : **TEB->PEB->PEB_LDR_DATA**) ;

- techniques de récupération des mots de passe : un malware peut embarquer avec lui des outils de récupération de mots de passe comme **[mimikatz]**, **[pwdump]** et autres. Comme il peut implémenter ses propres fonctions de capture de frappe clavier (**SetWindowsHookEx**, **GetAsyncKeyState**), ou de récupération des condensats de mot de passe (**lpGetPrimaryCredential**).

Cependant, l'analyse peut rapidement devenir très complexe en fonction de plusieurs facteurs :

- l'optimisation du compilateur : le type du compilateur et les options choisies au moment de la compilation impliquent la génération de certains motifs au niveau du binaire (insertion inline d'une fonction, propagation des constantes, déroulage et inversion de boucle, etc.) **[Compiler_OPTI]** ;
- l'obscurcissement du code : vise à rendre le désassemblage et la lecture du code difficiles voire erronés, en insérant du code inutile (*junk code*), du code auto-modifiant ou des **[prédicats obscurs]** ;
- l'obscurcissement du flux d'exécution : ces techniques interviennent au moment de l'exécution pour modifier le flux de cette dernière en se basant sur des mécanismes tels que les exceptions (**SEH**) ou les **TLS** (*Thread Local Storage*). En effet, un malware peut s'amuser à lever des exceptions lui permettant d'exécuter du code rendant la simple lecture du code statique inefficace ;
- le langage de développement : en effet, la complexité du code analysé peut rapidement croître selon la nature du langage de développement utilisé. Les applications C++ par exemple, se trouvent dotées de structures *RTTI* et *Vtables* **[MISC_RTTI]** assurant les relations d'héritage et rendant complexe la lecture du code (utilisation des pointeurs de fonctions : **call eax**), sans oublier les applications de type COM, ActiveX, BHO (*Browser Helper Object*), MFC (*Microsoft Foundation Classes*) ou les applications multi-threads.

Pour toutes ces tâches, le décompilateur **[HexRays]** d'IDA pro peut se révéler très utile.

6

En route vers l'antidote : fiche d'incident

La fin de notre processus d'analyse est marquée par la production d'un rapport. Ce dernier doit fournir une vue détaillée du malware, car il servira comme support de travail à la fois pour l'équipe de gestion d'incident et l'équipe d'analyse de malwares. Cette dernière utilisera son contenu pour alimenter sa base de capitalisation et ainsi mettre en évidence les liens qui existeraient entre les différentes campagnes.

Ce rapport doit mettre en évidence l'ensemble des IOC identifiés et peut être organisé comme suit :

- un résumé reprenant les principales fonctions du malware : indiquant si le malware a été développé spécifiquement pour la cible ;
- une indication sur le taux de couverture de l'analyse ;
- les méta-informations sur les binaires analysés : comprenant le nom du fichier, son empreinte (MD5/SHA1), etc. ;
- les moyens de communication utilisés par le malware : le protocole réseau utilisé (HTTP, FTP, etc.), l'adresse IP du serveur de contrôle, le nom de domaine, les commandes gérées par le malware, les valeurs utilisées par le malware pendant ses échanges (User-Agent, paramètres POST, valeur magique, etc.) ;
- les indicateurs de compromission : clés de registres, fichiers, Mutex, ports de communications Windows, etc. ;
- les moyens de persistance : tâches planifiées, services Windows, clés de registre, etc. ;
- la nature de l'information divulguée (mot de passe, documents, courriels, etc.) ainsi que le format utilisé pour l'exfiltration ;
- les mécanismes de protection (chiffrement, *packing*, anti-virus, auto-destruction, etc.) utilisés par le malware en soulignant le niveau de complexité de ces protections (absent, moyen, élevé) ;
- les algorithmes de chiffrement utilisés.

Conclusion : la guerre n'est jamais finie !

Hélas, les techniques utilisées par les malwares et les outils d'analyses ne cessent d'évoluer et la course pour l'armement ne fait qu'accélérer. L'analyste se doit d'être à l'affût des dernières techniques et adapter continuellement sa trousse à outils. De ce fait, l'industrialisation du processus d'analyse devient un besoin critique pour toutes les équipes de gestion d'incident. Toutefois, l'automatisation de certaines tâches comme l'analyse statique de code reste dans le domaine de la recherche. L'article met en lumière diverses « recettes » manuelles, mais qui restent nécessaires pour assurer la précision d'une analyse. ■

■ REMERCIEMENTS

Un grand merci à toute l'équipe AGI, particulièrement Raphaël, Axel et Sarah pour leurs relectures et conseils.

Retrouvez les références de cet article sur :

www.miscmag.com

MÉTHODOLOGIE D'ANALYSE DE VULNÉRABILITÉS ET PRODUCTION DE SIGNATURES DE DÉTECTION

Jean-Paul Roliers – Bureau Failles et Signature – bfs@ssi.gouv.fr



mots-clés : *VULNÉRABILITÉ / INTERNET EXPLORER / CVE-2014-0322 / SIGNATURE / DÉTECTION*

Cet article présente, à travers l'analyse de la vulnérabilité CVE-2014-0322 affectant Internet Explorer, le cheminement suivi pour parvenir à la détermination de signatures permettant de détecter les codes exploitant cette vulnérabilité.

1 Contexte

Dans le cadre de la réponse à incidents, les équipes chargées de l'analyse de la compromission peuvent être amenées à récupérer, au travers des fichiers présents sur le disque ou des données ayant transité par le réseau, le binaire responsable de l'entrée de l'attaquant sur le système.

Souvent, le code malveillant se présente sous la forme d'un document qui, lors de son interprétation, tentera d'exploiter une vulnérabilité présente dans le lecteur. Parmi les lecteurs les plus fréquemment visés, il convient de citer les navigateurs, les lecteurs PDF ainsi que les outils de traitement de texte.

En général, ces logiciels ont pour point commun l'utilisation de différents langages de scripting pour le traitement du format de fichier qu'ils interprètent.

Une fois qu'un échantillon a été obtenu, l'analyste en vulnérabilités peut alors démarrer son analyse dans le but de fournir plusieurs types de signatures : des signatures système (fichiers déposés, clefs de registre modifiées...), des signatures réseau (domaines contactés, fichiers téléchargés...) et des signatures « comportementales » (appels système spécifiques, dissémination de données en mémoire...).

2 Méthodologie d'analyse appliquée à l'exploit ciblant la vulnérabilité CVE-2014-0322

2.1 Généralités

L'utilisation de cette vulnérabilité dans le cadre d'attaques a été identifiée publiquement le 11 février 2014 par la société FireEye [1]. Un exploit était déployé sur un site dédié aux vétérans de guerre des États-Unis dans le cadre d'une attaque de type « point d'eau ».

Cette vulnérabilité vise le navigateur Microsoft Internet Explorer dans sa version 10 et a fait l'objet d'un correctif dans le bulletin mensuel de Microsoft MS14-012 du 11 mars 2014 [2] [3].

Le code d'exploitation analysé se présente sous la forme de trois fichiers : une application Flash incluse dans une page HTML (qui elle-même contient du code JavaScript) et une image.

Lors de l'étude d'une vulnérabilité, il est recommandé que l'analyste travaille sur un poste déconnecté :



en cas de compromission, seul le poste d'analyse est affecté et non l'ensemble du réseau auquel il est connecté.

2.2 Présentation de la vulnérabilité

2.2.1 Préparation d'une preuve de concept

Le code JavaScript déclenchant la vulnérabilité est similaire au code suivant disponible publiquement :

```
function dword2data(dword)
{
    var d = Number(dword).toString(16);
    while (d.length < 8)
        d = '0' + d;
    return unescape('%u' + d.substr(4, 8) + '%u' + d.substr(0, 4));
}

var g_arr = [];
function fun()
{
    var a = 0;
    var b = dword2data(0xdead0de);
    var c = 0x1a1b2000;

    for (a = 0; a < 0x250; ++a)
        g_arr[a] = document.createElement('div');

    while (b.length < 0x360)
    {
        if (b.length == (0x94 / 2))
            b += dword2data(c + 0x10 - 0x0c);
        else if (b.length == (0x98 / 2))
            b += dword2data(c + 0x14 - 0x08);
        [...]
    }
    [...]

    this.outerHTML = this.outerHTML

    CollectGarbage();
    var d = b.substring(0, (0x340 - 2) / 2);
    for (a = 0; a < 0x250; ++a)
        g_arr[a].title = d.substring(0, d.length);
}

function puIHa3()
{
    [...]
    b.onpropertychange = fun;
    var c = document.createElement('SELECT');
    c = b.appendChild(c);
}
```

Afin de découvrir l'emplacement du code responsable de la redirection du flot d'exécution, il est plus aisé de modifier l'exploit fonctionnel pour obtenir une

preuve de concept neutralisée. Dans notre cas, la fonction JavaScript **dword2data** est modifiée afin de ne retourner que la valeur **0x41414141**.

Après avoir attaché le debugger au processus Internet Explorer chargé du rendu des pages et en se connectant à la page contenant la preuve de concept, l'exception suivante est obtenue :

```
MSHTML!CMarkup::UpdateMarkupContentsVersion+0x16:
66cce4ee ff4010inc dword ptr [eax+10h] ds:0023:41414151=????????
```

L'exception est déclenchée par le déréférencement d'une adresse invalide, à savoir **0x41414151**, qui correspond au résultat de **EAX + 0x10**. Le registre EAX contient donc la valeur **0x41414141** obtenue suite à un appel à la fonction **dword2data** précédemment modifiée.

2.2.2 Analyse de l'exception

Le contenu du registre EAX provient d'un déréférencement antérieur du registre EDX. Le contenu de ce registre provient lui-même des arguments passés en paramètre à la fonction **CMarkup::UpdateMarkupContentsVersion** par la fonction parente **CMarkup::NotifyElementEnterTree**.

```
CMarkup::NotifyElementEnterTree
mov esi, [ebp+arg_0]
[...]
mov edx, esi
call CMarkup::UpdateMarkupContentsVersion

CMarkup::UpdateMarkupContentsVersion
[...]
mov eax, [edx+0ACh]
[...]
inc dword ptr [eax+10h]
```

En plaçant un point d'arrêt lors de l'appel à la fonction **CMarkup::NotifyElementEnterTree**, on découvre, grâce à la présence d'une **vtable**, que la variable contenue dans **arg_0** est un objet C++ de type **CMarkup**.

```
0:007> dps poi(poi(esp+4))
61324208 6142ba5d MSHTML!CMarkup::PrivateQueryInterface
6132420c 6133b8e4 MSHTML!CMarkup::PrivateAddRef
61324210 6133b8c1 MSHTML!CMarkup::PrivateRelease
[...]
```

En suivant la pile d'appels, on constate qu'il est vraisemblable que l'appel de cette fonction provienne du code JavaScript « **c = b.appendChild(c);** ».

```
0:007> kv
MSHTML!CMarkup::NotifyElementEnterTree
MSHTML!CMarkup::InsertSingleElement+0x169
MSHTML!CMarkup::InsertElementInternalNoInclusions+0x11d
MSHTML!CMarkup::InsertElementInternal+0x2e
MSHTML!CMarkup::InsertElement+0x9c
```



```
MSHTML!DOMNodeHelper+0x454
MSHTML!CElement::InsertBeforeHelper+0x92
MSHTML!CElement::InsertBeforeHelper+0xe5
MSHTML!CElement::InsertBefore+0x36
MSHTML!CElement::Var_appendChild+0xcb
MSHTML!CFastDOM::CNode::Trampoline_appendChild+0x55
jscript9!Js::JavascriptExternalFunction::ExternalFunctionThunk+0x185
```

En plaçant un point d'arrêt mémoire sur la variable **arg_0**, on détecte qu'une modification est effectuée lors de l'appel au destructeur de l'objet. Il semblerait que cet appel provienne du code JavaScript « **this.outerHTML = this.outerHTML** ».

```
0:007> kv
MSHTML!CMarkup::~CMarkup+0x16
MSHTML!CMarkup::`vector deleting destructor'+0xd
MSHTML!CBase::SubRelease+0x2e
MSHTML!CBase::PrivateRelease+0x7f
MSHTML!CMarkup::Release+0x2d
MSHTML!InjectHtmlStream+0x6f9
MSHTML!HandleHTMLInjection+0x82
MSHTML!CElement::InjectInternal+0x521
MSHTML!CElement::InjectTextOrHTML+0x1a4
MSHTML!CElement::put_outerHTML+0x1d
MSHTML!CFastDOM::CHTMLElement::Trampoline_Set_outerHTML+0x54
[...]
jscript9!JavascriptDispatch::InvokeEx+0x1e5
MSHTML!CBase::InvokeDispatchWithThis+0xb9
MSHTML!CBase::InvokeEvent+0x2a2
MSHTML!CBase::FireEvent+0x12e
MSHTML!CElement::FireEvent+0x25d
MSHTML!CElement::Fire_onpropertychange+0x7e
MSHTML!CElement::Fire_PropertyChangeHelper+0xfe
MSHTML!CElement::OnPropertyChange+0x743
MSHTML!CScriptElement::OnPropertyChange+0x16
MSHTML!BASICPROPPARAMS::SetStringProperty+0x4b5
MSHTML!CBase::put_StringHelper+0x5e
MSHTML!CScriptElement::SetPropertyHelper+0x19
MSHTML!CScriptElement::OnTextChanged+0x53
MSHTML!CElement::HandleTextChanged+0x37
MSHTML!CMarkup::NotifyElementEnterTree+0x1e4
```

En reprenant l'exécution du programme, un nouvel accès au point d'arrêt mémoire est déclenché, cette fois-ci par la fonction **memcpy**, qui va copier vers la zone mémoire précédemment libérée la valeur **0x41414141**.

```
0:007> kv
msvcrt!memcpy+0x5a
MSHTML!CAttrArray::Set+0x3c2
MSHTML!CAttrArray::Set+0x37
MSHTML!CAttrArray::SetString+0x41
MSHTML!BASICPROPPARAMS::SetString+0x30
MSHTML!BASICPROPPARAMS::SetStringProperty+0x48a
MSHTML!CBase::put_StringHelper+0x5e
MSHTML!CFastDOM::CHTMLElement::Trampoline_Set_title+0x76
jscript9!Js::JavascriptExternalFunction::ExternalFunctionThunk+0x185
[...]
MSHTML!CMarkup::NotifyElementEnterTree+0x1e4
```

Enfin, en continuant l'exécution, l'exception vue précédemment est levée.

À partir de cette trace, plusieurs conclusions peuvent être tirées sur cette vulnérabilité :

- il s'agit d'une vulnérabilité de type *User-After-Free* : après libération de l'objet **CMarkup**, ce dernier est réutilisé ;
- les éléments du code JavaScript qui déclenchent la vulnérabilité sont :
 - ↳ **b.appendChild(c)** qui provoque l'appel initial au chemin de code vulnérable ;
 - ↳ **this.outerHTML = this.outerHTML** qui provoque la libération de l'objet **CMarkup** ;
 - ↳ **g_arr[a].title = d.substring(0, d.length)** qui réécrit la zone mémoire précédemment utilisée par l'objet **CMarkup**.

2.3 Exploitation de la vulnérabilité

Le code de la fonction vulnérable (**CMarkup::UpdateMarkupContentsVersion**) ne présente pas de code permettant à l'attaquant une redirection « directe » du flot d'exécution (comme un CALL sur la valeur d'un registre). Cette vulnérabilité fournit « seulement » une primitive d'incrémentement sur un objet contrôlé par l'attaquant.

Afin de comprendre comment cette primitive est utilisée, il est nécessaire de décompiler le code flash [4].

2.3.1 Fonctionnement du code Flash

Le code Flash est contenu dans une seule classe ActionScript appelée **Tope**. Le constructeur de l'objet se charge de télécharger une image nommée « Erido.jpg ». Une fois ce téléchargement terminé, une nouvelle fonction est appelée.

Le but de cette fonction est de disséminer dans le tas du processus une multitude d'objets **Vector** de deux types : les premiers contiendront des entiers tandis que les seconds contiendront une référence vers un objet de type Sound précédemment instancié.

En ActionScript, les objets **Vector** peuvent être assimilés à des tableaux d'objets dont la taille est limitée. En observant ces objets en mémoire, l'agencement suivant apparaît :

Vector_len	Vector_vtable	Vector_element_1	...	Vector_element_n
4 octets	4 octets	4 octets	...	4 octets

Une fois le tas rempli de ces objets, la fonction JavaScript **puIHa3** est appelée deux fois et un timer est déclenché afin d'appeler une fonction Flash une fois ce dernier écoulé.

Il est maintenant nécessaire de lancer l'exploit en prenant soin de positionner un point d'arrêt sur la fonction **CMarkup::UpdateMarkupContentsVersion** qui, précédemment, provoquait le plantage d'Internet Explorer.

Lors du déclenchement du point d'arrêt, la mémoire est organisée de la manière suivante :

```
MSHTMLCMarkup::UpdateMarkupContentsVersion:
6941e4d8 8b427cmov eax,dword ptr [edx+7Ch]
ds:0023:08c37f9c=1a1b1ff0
0:007> dd edx L 400
08c37f20 deadc0de 1a1b1ff0 1a1b1ff0 1a1b1ff0
08c37f30 1a1b1ff0 1a1b1ff0 1a1b1ff0 1a1b1ff0
08c37f40 1a1b1ff0 1a1b1ff0 1a1b1ff0 1a1b1ff0
[...]
0:007> dd 1a1b1ff0
1a1b1ff0 00000000 00000000 00000000 00000000
1a1b2000 000003fe 080d3000 deadbee1 00000000
1a1b2010 1a1b2000 1a1b2000 00000000 00000000
1a1b2020 00000000 00000000 00000000 00000000
[...]
```

Le registre EAX prend alors pour valeur une adresse du tas (**0x1a1b1ff0**) qui elle-même est positionnée 0x10 octets avant le début d'un objet de type **Vector**. Suite aux deux appels à la fonction vulnérable, le contenu du tas devient :

```
0:007> dd 1a1b1ff0
1a1b1ff0 00000000 00000000 00000000 00000000
1a1b2000 00000400 080d3000 deadbee1 00000000
1a1b2010 1a1b2000 1a1b2000 00000000 00000000
1a1b2020 00000000 00000000 00000000 00000000
[...]
```

Après cette modification, et une fois le timer écoulé, le code ActionScript reprend la main et va alors parcourir l'ensemble des **Vector** d'entiers, créés précédemment, à la recherche d'un objet **Vector** dont la taille sera supérieure à celle d'origine.

Une fois cet élément localisé, le code va accéder à la dernière entrée du tableau et lui affecter la valeur **0x3fffffff0**. Cette entrée, positionnée à l'extérieur du **Vector** à cause de l'agrandissement de sa taille, a pour but d'écraser la taille du **Vector** suivant avec une grande valeur.

Le code ActionScript va ainsi pouvoir (en utilisant le **Vector** dont la taille a été passée à **0x3fffffff0**) accéder, en lecture et en écriture, à une large portion de la mémoire du processus.

Tout d'abord, des objets vont être recherchés en mémoire afin d'obtenir les éléments nécessaires à la construction d'une chaîne ROP. On retrouve notamment la recherche d'un gadget de type **stack pivot** ainsi que l'adresse de la fonction **ZwProtectVirtualMemory** et l'adresse de l'objet Sound en mémoire. Il est à noter que cette résolution dynamique des objets permet de contourner l'ASLR.

Une large zone mémoire va ensuite être lue et stockée par le code ActionScript afin d'effectuer une sauvegarde de la zone du tas qui sera modifiée, dans le but de la restaurer une fois le code d'exploitation terminé.

Ensuite la possibilité d'écriture offerte par l'objet **Vector** modifié va être utilisée afin de construire, sur le tas, la chaîne ROP et une première partie du *shellcode*. Cette partie sera ensuite complétée par des données issues de l'image Erido.jpg téléchargée précédemment. Une analyse complète du format de cette image est disponible à l'adresse suivante [5].

Enfin, l'adresse de la **vtable** de l'objet Sound est modifiée afin de pointer vers une nouvelle **vtable**, forgée par l'attaquant, située dans la zone du tas sous son contrôle.

Finalement le transfert du flot d'exécution est fait grâce à l'appel à la fonction **toString** de l'objet Sound.

Lorsqu'un point d'arrêt mémoire est positionné sur l'emplacement du **stack pivot** (xchg eax, esp ; ret;), il est possible de retrouver le code qui tente d'accéder en lecture à cette instruction :

```
mov eax, [ecx]
mov edx, [eax+70h]
call edx
```

ECX étant un pointeur vers l'adresse du début de la chaîne ROP, EAX est donc chargé avec l'adresse de la chaîne. Après l'exécution du **stack pivot**, ESP se retrouve donc en tout début de la chaîne ROP.

La disposition mémoire de la chaîne ROP est la suivante :

0x1a1b3100	Adresse de ZwProtectVirtualMemory	
0x1a1b3104	Adresse de retour (0x1a1b311c)	+- -----+
0x1a1b3108	Argument 1	
[...]	[...]	
0x1a1b311c	Shellcode (partie 1)	<----- -----+
[...]	[...]	
0x1a1b3170	Adresse du stack pivot	
0x1a1b3174	Shellcode (partie 2)	

L'appel à **ZwProtectVirtualMemory** avec les arguments pointant vers le tas permet de rendre ce dernier exécutable. Ainsi, lorsque le code saute sur la première partie du *shellcode*, celui-ci s'exécute sans problème. Cette partie se charge de sauvegarder certains éléments et de restaurer la véritable adresse de la **vtable** de l'objet Sound. Ensuite, le code saute sur la deuxième partie du *shellcode*.

2.4 Analyse du shellcode

Toujours avec un point d'arrêt mémoire sur l'emplacement du **stack pivot**, on effectue un dump du *shellcode* pour ensuite l'émuler à l'aide de **scdbg** [6].

```
scdbg.exe /f sc_raw
Loaded 40001 bytes from file sc_raw
Detected straight hex encoding input format converting...
Initialization Complete..
Max Steps: 2000000
Using base offset: 0x401000

40110f LoadLibraryA(shell32)
40112d LoadLibraryA(user32)
401148 GetTempPathA(len=100, buf=12fb00) = 20
401182 GetTempPathA(len=100, buf=12fc00) = 20
4011cd CreateFileA(C:\Users\IEUser\AppData\Local\Temp\sqlrenew.txt) = 4
401243 WriteFile(h=4, buf=401475, len=400, lpw=12fa74, lap=0) = 1
    Hiding repetitive WriteFile calls
401269 CloseHandle(4)
4012a7 CreateFileA(C:\Users\IEUser\AppData\Local\Temp\stream.exe) = 8
4012fb WriteFile(h=8, buf=40b875, len=400, lpw=12fa74, lap=0) = 1
    Hiding repetitive WriteFile calls
401321 CloseHandle(8)
40135e LoadLibraryA(C:\Users\IEUser\AppData\Local\Temp\sqlrenew.txt)
    Unknown Dll - Not implemented by libemu
401386 error accessing 0x1a1b2024 not mapped
[...]
```

À l'aide de ces informations, on peut voir, à moindre coût, que le *shellcode* dépose deux fichiers sur le disque, puis charge le premier en mémoire en tant que bibliothèque.

Une analyse plus poussée de ce *shellcode* permet de repérer des comportements spécifiques (que l'on a peu de chance de retrouver dans un binaire légitime) :

- résolution de fonctions utilisées par le *shellcode* via l'EAT ;
- boucle XOR avec la valeur **0x95** avant l'appel à **WriteFile** ;
- avant l'appel à une fonction, détection de la présence de routines d'interception (JMP ou CALL à la place du prologue) et si c'est le cas, contournement de ces dernières.

2.5 Création de signatures

Lors de la création de signatures, il est important de garder en tête que ces dernières ne doivent pas générer trop de faux positifs afin de rester pertinentes et exploitables par les équipes de supervision. Il est donc nécessaire de se concentrer sur des éléments qui caractérisent l'exploitation de la vulnérabilité et qui ne se retrouvent pas dans des fichiers ou du trafic légitime.

De plus, il est bon de pouvoir disposer d'un jeu de données légitime varié afin de vérifier que les signatures qui ont été créées n'engendrent pas de faux positifs.

2.5.1 Signatures sur les données en transit par le réseau

Code source JavaScript

Comme vu précédemment, la succession d'appels effectués par le code JavaScript mène à une exception dans le code de **mshtml.dll**. La présence de ces fonctions dans un document HTML peut permettre de suspecter ce dernier. On pourra par exemple créer une règle Yara détectant la présence de ces appels dans un fichier HTML :

```
rule CVE_2014_0322
{
    strings:
        $inner1 = "this.outerHTML"
        $inner2 = ".appendChild("
        $inner3 = ".title"
    condition:
        $inner1 and $inner2 and $inner3
}
```

Dans le cas où le code source JavaScript est obfusqué, il est possible d'utiliser un désobfuscateur ou de baser les règles sur un arbre syntaxique abstrait. Cependant l'élaboration de ces outils est coûteuse en temps de développement pour des résultats qui ne sont pas toujours au rendez-vous.

Code source Flash

Des règles de détections peuvent être implémentées sur le code source Action Script. Il est cependant nécessaire de disposer d'outils d'analyse permettant la décompression/décompilation du code.

Plusieurs éléments du code sont discriminants. La boucle de recherche des bibliothèques chargées en mémoire ou le code responsable de la création de la chaîne ROP et du *shellcode* (dont la plupart des valeurs sont statiques) sont de bons candidats. Les valeurs fixes correspondant aux opcodes des actions spécifiques du *shellcode* vues précédemment sont d'autant plus intéressantes.

Image téléchargée

Le format spécifique utilisé pour embarquer des binaires dans le fichier image peut permettre de détecter la présence des en-têtes PE dans le fichier. Il est nécessaire pour cela d'implémenter un moteur qui se chargera de tenter de « déchiffrer » (XOR 0x95) le fichier à l'emplacement où les binaires sont attendus.



Il est à noter que cette méthode peut être étendue à d'autres types de fichiers, et ce sans connaître la clef de déchiffrement (en utilisant une attaque par force brute) afin de rechercher les en-têtes PE comme le fait par exemple l'outil XORSearch [7].

2.5.2 Signatures sur les journaux réseau

L'URL de récupération du fichier Erido.jpg est dépendante de l'emplacement du fichier Flash. L'apparition des fichiers Tope.swf et Erido.jpg provenant du même hôte sur le fichier journal d'un équipement réseau peut signaler une tentative d'exploitation d'un poste du réseau.

2.5.3 Signatures sur les fichiers déposés

Une fois le flot d'exécution transféré, le shellcode dépose deux fichiers. Il est possible de créer une signature OpenIOC qui détecterait la présence de ces fichiers sur le disque lors d'une investigation numérique en précisant le nom ainsi que le condensat de ces fichiers.

2.5.4 Signature sur l'exécution du processus

Durant l'exécution du processus dans un environnement de type bac à sable, il est possible d'intercepter les appels aux fonctions système. Ces appels sont le plus souvent légitimes. Cependant, appelés avec certains paramètres, ils peuvent devenir discriminants.

L'allocation consécutive d'une multitude d'éléments de la même taille sur le tas et son remplissage par une valeur spécifique, l'utilisation de la fonction **ZwProtectVirtualMemory** en demandant l'autorisation d'accès en exécution sur une zone sur le tas qui, de plus, contient l'adresse de retour de la fonction et l'utilisation de la fonction **LoadLibrary** avec en paramètre un fichier texte sont autant d'appels de fonctions qui peuvent trahir un comportement malveillant.

De plus l'utilisation d'EMET [8] dans un environnement de type bac à sable peut permettre de détecter un ensemble de comportements suspects sans avoir à fournir, au préalable, de signatures spécifiques.

2.5.5 Limites des signatures

Le problème principal inhérent aux signatures est qu'il est toujours possible pour l'attaquant de modifier

le comportement de son code afin de parvenir à ne plus être détecté.

Pour chaque signature de détection présentée, une méthode de contournement peut être mise en place : le code JavaScript renvoyé au navigateur peut être obfusqué, le nom des fichiers et le condensat des fichiers Flash et JPG peuvent être changés, le *shellcode* et sa charge utile peuvent être modifiés, etc.

Cependant, la difficulté de la mise en place de ces modifications n'est pas la même : s'il est aisé d'obfusquer le code JavaScript transmis ou de changer la charge utile déposée sur le disque, d'autres opérations, telle que le changement de la structure de l'image contenant les binaires, peuvent être plus compliquées notamment pour un attaquant n'ayant pas développé l'exploit, car elles nécessitent une action sur plusieurs éléments : outil de génération de l'image, shellcode, fichier Flash.

Conclusion

La création de signatures dans le cadre de l'analyse de vulnérabilités nécessite une compréhension poussée du code d'exploitation depuis l'interprétation du fichier corrompu à l'exécution de code arbitraire. Cette analyse est nécessaire afin de comprendre la cause de l'exécution de code et le comportement du shellcode afin de pouvoir en extraire des signatures. Ces dernières doivent être un compromis entre des signatures les plus génériques permettant de détecter des modifications du code d'exploitation, mais générant du bruit (faux positifs), et des signatures spécifiques ne ciblant qu'un code d'exploitation particulier, mais sans générer d'erreur. ■

■ DOCUMENTATION

- [1] <http://www.fireeye.com/blog/technical/cyber-exploits/2014/02/operation-snowman-deputydog-actor-compromises-us-veterans-of-foreign-wars-website.html>
- [2] <https://technet.microsoft.com/library/security/ms14-012>
- [3] <http://cert.ssi.gouv.fr/site/CERTFR-2014-AVI-118/>
- [4] <http://www.free-decompiler.com/flash/>
- [5] <http://labs.bromium.com/2014/02/25/dissecting-the-newest-ie10-0-day-exploit-cve-2014-0322/>
- [6] <http://sandsprite.com/blogs/index.php?uid=7&pid=152>
- [7] <http://blog.didierstevens.com/programs/xorsearch/>
- [8] <http://technet.microsoft.com/fr-fr/security/jj653751>

NOUVEAU ! NOUVEAU ! NOUVEAU ! NOUVEAU !

ABONNEZ-VOUS!

DÉCOUVREZ NOS NOUVELLES OFFRES !

LE CLASSIQUE !

VERSION

PAPIER



Retrouvez tous les 2 mois votre bimestriel favori en papier dans votre boîte à lettres !



Envie de lire tous les 2 mois votre magazine sur votre tablette ou votre ordinateur ? Voici l'abonnement PDF !

Attention, les offres mentionnées dans la page suivante sont réservées aux particuliers, professionnels n'hésitez pas à nous contacter à : abopro@ed-diamond.com

PDF



L'ÉLECTRONIQUE !

VERSION

PDF

LE NOUVEAU SERVICE !

ACCÈS À LA

BASE DOCUMENTAIRE



L'accès à la base documentaire en ligne de MISC et de ses Hors-séries vous permettra d'effectuer des recherches dans la majorité des articles parus, qui seront disponibles avec un décalage de 6 mois après leur parution en magazine.

Attention, les offres mentionnées dans la page suivante sont réservées aux particuliers, professionnels n'hésitez pas à nous contacter à : abopro@ed-diamond.com



boutique.ed-diamond.com

Découvrez une sélection d'offres page suivante



RETROUVEZ TOUTES LES AUTRES OFFRES D'ABONNEMENTS SUR : **boutique.ed-diamond.com !**



Merci de cocher ci-dessous l'offre souhaitée...

PAPIER	TARIF NORMAL	ÉCONOMIE	TARIF SPÉCIAL*
☐ MISC 6 ^{N°}	53,40	11,40	42,-€

PDF**	TARIF NORMAL	ÉCONOMIE	TARIF SPÉCIAL*
☐ MISC 6 ^{N°}	53,40	11,40	42,-€

BASE DOCUMENTAIRE**	TARIF SPÉCIAL*
☐ ACCÈS 1 AN	59,-€

PAPIER + PDF**	TARIF NORMAL	ÉCONOMIE	TARIF SPÉCIAL*
☐ MISC 6 ^{N°} + MISC 6 ^{N°}	106,80	44,80	62,-€

PAPIER + BASE DOCU**	TARIF NORMAL	ÉCONOMIE	TARIF SPÉCIAL*
☐ MISC 6 ^{N°} + ACCÈS 1 AN	112,40	23,40	89,-€

PAPIER + PDF** + BASE DOCUMENTAIRE**	TARIF NORMAL	ÉCONOMIE	TARIF SPÉCIAL*
☐ MISC 6 ^{N°} + MISC 6 ^{N°} + ACCÈS 1 AN	165,80	54,80	111,-€



Merci de cocher ci-dessous l'offre souhaitée...

PAPIER	TARIF NORMAL	ÉCONOMIE	TARIF SPÉCIAL*
☐ MISC 6 ^{N°} + MISC HS 2 ^{N°}	71,40	17,40	54,-€

PDF**	TARIF NORMAL	ÉCONOMIE	TARIF SPÉCIAL*
☐ MISC 6 ^{N°} + MISC HS 2 ^{N°}	71,40	17,40	54,-€

PAPIER + PDF**	TARIF NORMAL	ÉCONOMIE	TARIF SPÉCIAL*
☐ MISC 6 ^{N°} + MISC HS 2 ^{N°} + MISC 6 ^{N°} + MISC HS 2 ^{N°}	142,80	58,80	84,-€

PAPIER + BASE DOCU**	TARIF NORMAL	ÉCONOMIE	TARIF SPÉCIAL*
☐ MISC 6 ^{N°} + MISC HS 2 ^{N°} + ACCÈS 1 AN	130,40	32,40	98,-€

PAPIER + PDF** + BASE DOCUMENTAIRE**	TARIF NORMAL	ÉCONOMIE	TARIF SPÉCIAL*
☐ MISC 6 ^{N°} + MISC HS 2 ^{N°} + MISC 6 ^{N°} + MISC HS 2 ^{N°} + ACCÈS 1 AN	201,80	68,80	133,-€

Vous souhaitez également vous abonner à :



Rendez-vous sur notre boutique pour découvrir toutes nos offres de couplages !
boutique.ed-diamond.com

*Prix TTC France Métro. Pour les tarifs hors France Métro et étranger, veuillez consulter notre boutique en ligne !

**Offre réservée aux particuliers, professionnels contactez-nous par mail : abopro@ed-diamond.com

Voici mes coordonnées postales :

Société :	
Nom :	
Prénom :	
Adresse :	
Code Postal :	
Ville :	
Pays :	
Téléphone :	
E-mail :	

☐ Je souhaite recevoir les offres promotionnelles et newsletters des Éditions Diamond.

☐ Je souhaite recevoir les offres promotionnelles des partenaires des Éditions Diamond.

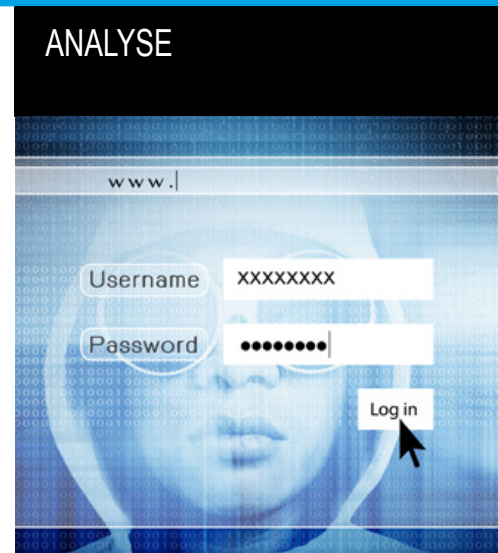


Édité par Les Éditions Diamond
Service des Abonnements
B.P. 20142 - 67603 Sélestat Cedex
Tél. : + 33 (0) 3 67 10 00 20
Fax : + 33 (0) 3 67 10 00 21

Vos remarques :

RECHERCHE D'INDICES DE COMPROMISSION SUR IOS

Arnaud Malard (ANSSI) et Mathieu Renard (ANSSI)



mots-clés : APPLE / IOS / INVESTIGATION NUMÉRIQUE / IOC / MALWARES

Cet article est une introduction à l'investigation numérique de terminaux Apple iOS. L'objectif est d'infirmer ou confirmer la compromission du terminal par un programme malveillant. Dans les différents exemples, nous partons du principe que le terminal à analyser est fourni volontairement par la victime et donc que l'éventuel mot de passe protégeant le terminal est connu. D'autre part, les thèmes relatifs à l'analyse des éléments de la chaîne de démarrage du terminal (BootRom, LLB, iBoot, etc.) ainsi que le code exécuté sur la puce radio (Baseband) ne seront pas abordés, leur complexité d'analyse nécessitant un article à eux seuls.

1 Architecture en bref

Afin de donner au lecteur tous les éléments nécessaires à la compréhension des contraintes liées à la réalisation d'une investigation numérique sur un terminal Apple, ce chapitre présente succinctement l'architecture d'un système iOS.

1.1 Architecture & SoC applicatif

L'architecture matérielle des terminaux Apple repose principalement sur leur SoC (*System on Chip*). Ce SoC, qualifié à tort de processeur, ne cesse d'évoluer au rythme des versions des terminaux. Parmi les fonctionnalités les plus importantes de ce système intégré, il faut noter la présence d'un coprocesseur cryptographique utilisé pour les opérations de chiffrement et de déchiffrement des données stockées sur la mémoire NAND (flash). Ce cryptoprocèsseur étant placé en coupure entre la mémoire RAM et la NAND, tout échange de données entre lui et la RAM est chiffré. Autre point important, le cryptoprocèsseur renferme deux clés de chiffrement (les clés sont gravées dans le silicium du SoC) :

- la clé GID, propre à chaque famille d'équipements, permet de déchiffrer le contenu des éléments composant les images de mise à jour ;
- la clé UID, propre à chaque terminal, est utilisée pour le chiffrement du système et des fichiers sur le terminal.

1.2 Mécanismes de sécurité

Apple s'attache à réduire la surface d'attaque de ses équipements à chaque nouvelle version (matérielle et logicielle). Ainsi, un grand nombre de mécanismes de sécurité a été intégré au système. Parmi les plus importants, il faut noter la chaîne de démarrage de confiance basée sur l'implantation d'une racine de confiance dans une mémoire non réinscriptible de type ROM (*BootROM*), les fonctionnalités de protection des données utilisateur et les fonctionnalités permettant de garantir l'intégrité du système en fonctionnement. Cette dernière fonctionnalité intègre notamment des mécanismes :

- de restriction des privilèges ;
- de signature de code et de contrôle d'intégrité des binaires ;
- d'isolation des applications ;



- de restriction de l'exécution de code malveillant :
 - ↳ activation du bit *eXecute Never* (XN),
 - ↳ activation de l'*Address Space Layout Randomization* (ASLR) pour les espaces mémoire utilisateur et noyau,
 - ↳ activation de la *Position-independent executable* (PIE),
 - ↳ ajout de canaris sur la pile et le tas.

Ces différents mécanismes de sécurité s'appliquent à toutes les applications iOS, y compris aux outils d'investigation numérique. Ceci confère aux analyses d'un terminal Apple une grande complexité.

1.3 Protection des données

Afin d'interdire toute modification du système d'exploitation, Apple a choisi d'organiser le système de fichiers en deux partitions. La première partition HFS+ appelée « partition SYSTEM » par la suite, montée en lecture seule sur un système intègre, est utilisée pour stocker les données du système. Elle contient notamment le *kernelcache* (noyau du système et ses extensions), ainsi que l'ensemble des services, programmes et bibliothèques nécessaires au fonctionnement d'iOS.

La seconde partition HFS+, appelée « partition DATA » dans la suite de cet article, est réservée pour le stockage des données utilisateur. Celle-ci est donc accessible en lecture et écriture. Elle contient notamment l'ensemble des applications, leurs préférences et les données associées.

Comme la plupart des systèmes embarqués, ces partitions sont stockées sur la mémoire flash NAND du terminal, intégralement chiffrée avec une clé de volume. Celle-ci est elle-même chiffrée à l'aide d'une clé de chiffrement qui n'existe qu'en mémoire noyau et provenant d'une dérivation de la clé UID. Il est donc nécessaire d'avoir accès au processeur cryptographique pour pouvoir obtenir la clé de volume. Cette implémentation garantit la confidentialité des données lorsque l'équipement est hors tension et rend inutile toute extraction physique des données (démontage et lecture de la mémoire flash).

En ayant connaissance de la clé de volume, il est possible d'accéder au contenu des deux partitions. Toutefois, dans le but de restreindre l'accès aux fichiers de l'utilisateur et du système lors du fonctionnement, Apple a mis en place différentes couches de chiffrement additionnelles auxquelles sont associées différentes classes de protection. Il est donc nécessaire d'obtenir les clés de chiffrement associées à chacune de ces classes pour extraire les données de chacune des partitions.

Plus d'explications concernant les moyens complexes de chiffrement mis en œuvre par Apple sont disponibles dans les actes du **[SSTIC 2012]**.

2 Acquisition

L'acquisition de données est l'étape préalable à toute investigation numérique. Sur un terminal iOS, cette étape peut se révéler complexe du fait des limitations imposées par le système. Ce chapitre présente les différentes méthodes permettant l'extraction des données du terminal Apple. Le lecteur intéressé est invité à consulter l'article intitulé *Investigation numérique & terminaux Apple iOS : Acquisition de données* présenté lors de la conférence **[SSTIC 2014]**.

2.1 Précautions d'usage

Afin d'éviter toute communication avec l'environnement extérieur, il est recommandé de démarrer le terminal dans un environnement hors de portée de tout réseau mobile. Si l'analyste a accès à une cage de Faraday (ou un sac d'isolation électromagnétique), le terminal peut y être démarré dans le but d'identifier la version exacte du système installé (**Réglages > Général > Informations > Version**) et de réaliser les opérations d'acquisition.

2.2 Acquisition physique

Contrairement à l'acquisition logique, l'acquisition physique de données repose sur l'accès direct aux données, sans lancer la séquence de démarrage du terminal, autrement dit sans interaction avec le système d'exploitation. Or, comme précédemment évoqué, l'intégralité des données stockées sur la mémoire flash d'un terminal Apple est chiffrée. Sans connaissance des clés de chiffrement protégeant ces données, il est inutile de procéder à l'acquisition physique.

Publié en 2010 par George Hotzk alias GeoHot, le code d'exploitation **[LIMERAIN]** permet de démarrer un système alternatif personnalisé en exploitant une vulnérabilité dans le code immuable du BootROM des terminaux iPhone 2G/3G/3Gs/4 et iPad 1.

À partir de ces travaux, les chercheurs de la société « Sogeti ESEC » ont développé et mis à disposition une suite d'outils **[DATA-PROTECTION]** permettant de créer et de démarrer un système alternatif sur un terminal vulnérable. Cette technique est également utilisée par le boîtier d'acquisition **[UFED]**. Ainsi, depuis le système alternatif, il devient possible de réaliser une **image brute** (copie bit à bit) du contenu de la mémoire flash. Cette opération peut être réalisée à l'aide du script **dump_data_partition.sh** (à modifier pour extraire la partition SYSTEM).

Alors que l'image de la partition SYSTEM extraite n'est plus chiffrée (déchiffrement au démarrage du terminal), celle de la partition DATA (format **.dmg**) doit être déchiffrée pour être exploitable. Le script



emf_decrypter.py permet de réaliser cette opération en se basant notamment sur les clés dérivées de la clé UID et le trousseau d'accès (Keychain) précédemment extraits (via **dump_data_partition.sh**).

Le montage des partitions se réalise de la manière suivante (sous Mac OS X) :

```
$ hdiutil attach -readonly -imagekey diskimage-  
class=CRawDiskImage system_20140630-1650.dmg  
$ hdiutil attach -readonly data_20140630-1144.dmg
```

Les données de la partition DATA (accessible via **/private/var/** sur un terminal démarré) sont alors accessibles :

```
...  
0 drwxr-xr-x 10 user staff 374B 28 oct 2006 log/  
0 drwxr-xr-x 6 user staff 272B 26 déc 2007 logs/  
0 drwxrwxrwx 6 user staff 238B 25 jui 16:42 mobile/  
...
```

Ainsi que celles de la partition SYSTEM :

```
16 lrwxr-xr-x 1 user staff 23B 23 jui01:52 Applications@ -> /var/  
stash/Applications  
0 drwxrwxr-x 2 user staff 68B 17 déc 2012 Developer/  
0 drwxrwxr-x 16 user staff 748B 25 jui 15:16 Library/  
0 drwxr-xr-x 3 user staff 102B 21 sep 2012 System/  
16 lrwxr-xr-x 1 user staff 11B 30 jui 17:41 User@ -> /var/mobile  
...
```

2.3 Acquisition logique

À la date de rédaction de cet article (juillet 2014), aucune vulnérabilité publique n'affecte la chaîne de démarrage des terminaux 4s et supérieurs. Il est donc nécessaire d'avoir recours à d'autres méthodes d'acquisition de données. Ainsi deux techniques d'acquisition logique ont été présentées lors de l'édition 2014 de la conférence SSTIC **[SSTIC 2014]** par l'un des auteurs de cet article.

2.3.1 Acquisition logique sans altération du système

Il est possible d'extraire un grand nombre d'informations sans altération du système (sans *Jailbreak*) en s'appuyant sur les services normalement utilisés par les outils Apple (iTunes, par exemple).

Principaux services connus pour extraire des données système

La bibliothèque **[LIBIMOBILEDEVICE]**, qui implémente les différents protocoles nécessaires à l'utilisation des services accessibles (liste disponible dans **/System/Library/Lockdown/Services.plist**), permet d'extraire des données pertinentes dans le cadre d'une investigation numérique :

- **[DEVICEINFO]** utilise le service **com.apple.mobile.lockdown** pour obtenir les informations sur le terminal (adresses MAC, version du système, du *Baseband*, du matériel, etc.) ;
- **[DEVICEDATE]** permet la récupération de l'heure courante sur le terminal et donc la timezone ;
- **[DEVICEINSTALLER]**, basé sur le service **com.apple.mobile.installation_proxy**, permet entre autres de récupérer la liste des applications installées ainsi que leurs identifiants ;
- **[IFUSE]** utilise différents services : **com.apple.afc**, **com.apple.afc2** et **com.apple.mobile.house_arrest**. Cet outil permet d'accéder au contenu du répertoire **/var/mobile/Media.ifuse** permet également d'obtenir le contenu du répertoire d'une application tierce en spécifiant les paramètres « **--documents APPID** » (récupération des documents de l'application correspondante à l'identifiant spécifié) et « **--container APPID** » (récupération des fichiers des applications). En utilisant l'option « **--root** » il devient possible d'accéder à l'intégralité du système d'un terminal *jailbreaké* sur lequel le service **com.apple.afc2** est installé ;
- **[DEVICEPROVISION]** permet d'obtenir la liste des profils de développement installés sur le terminal via le service **com.apple.misagent** ;
- **[DEVICSYSLOG]** utilise le service **com.apple.syslog_relay** et permet d'obtenir en temps réel les journaux du terminal ;
- **[DEVICECRASHREPORT]** s'appuie sur le service **com.apple.crashreportmover** et autorise la récupération des rapports d'erreur ;
- **[DEVICEBACKUP]** & **[DEVICEBACKUP2]** utilisent les services **com.apple.mobile.backup** et **com.apple.mobile.backup2**, respectivement en charge de la sauvegarde iTunes et de la restauration du contenu du terminal. Une sauvegarde iTunes de terminal Apple contient les données suivantes :
 - ↳ GSM (historique des appels, SMS) ;
 - ↳ organisation (contacts, calendrier, notes) ;
 - ↳ média (photos, vidéos, musique, enregistrements) ;
 - ↳ web (historique de navigation Safari) ;
 - ↳ applications (liste des applications, préférences, cookies, contenu personnalisé) ;
 - ↳ fichiers de configuration **.plist** chiffrés (donc inexploitable).

Dans le cadre d'une recherche de marqueurs de compromission, les données extraites à travers ce type de sauvegarde n'ont pas de réel intérêt. En effet, des données pertinentes telles que les moyens de persistance (services, agents, etc.), les binaires des applications, les fichiers de journalisation ou les fichiers spéciaux ne sont pas accessibles. De plus, les dates des fichiers de la sauvegarde iTunes ne sont pas conservées et il n'est donc pas possible de générer une ligne de temps.

Service com.apple.mobile.file_relay

Du point de vue d'une investigation numérique, le service **com.apple.mobile.file_relay** est la meilleure source d'information. En effet, ce service est dédié à l'extraction d'informations. Les premières versions d'iOS supportent un nombre limité de sources de données, notamment :

- *SystemConfiguration* : permet d'obtenir la liste des SSID des réseaux Wi-Fi connus et la dernière fois qu'ils ont été rejoints ainsi que divers paramètres de configuration comprenant la configuration des interfaces réseau. Ces informations peuvent potentiellement être utilisées pour identifier le lieu d'une compromission ;
- *UserDatabases* : bases de données utilisateur (SMS, contacts...);
- *CrashReporter* : rapports d'erreurs ;
- *Caches* : liste détaillée des applications présentes sur le système et des variables d'environnement pour chacune des applications, etc. ;
- *Tmp* : contenu du répertoire **/tmp**.

Les nouvelles versions d'iOS (iOS 7) supportent un plus grand nombre de sources de données.

Parmi ces nouvelles sources de données, il faut souligner la source *HFSMeta*, qui permet de télécharger une image complète des métadonnées *HFS+* enregistrées sur la mémoire flash de l'appareil (depuis iOS7). L'image ainsi extraite contient toutes les informations relatives aux fichiers stockés sur le système (horodatage, noms des fichiers, taille, etc.). Cette source d'information vise à remplacer la source *VARFS* qui contenait un système de fichiers virtuel au format *statvfs*.

La source de données *Lockdown* permet d'obtenir une copie des informations générées lors de l'appairage (certificat/*escrowbag*). Elles peuvent être utilisées pour déterminer la liste des équipements sur lesquels le terminal a été synchronisé ou appairé. L'archive obtenue contient également une copie des informations générées lors de l'enregistrement et de l'activation du terminal. Ces données peuvent être utilisées pour déterminer le moment où le dispositif a été activé ou restauré pour la dernière fois. Les informations obtenues permettent également de déterminer d'autres paramètres de l'appareil tels que l'état des fonctionnalités de développement, de chiffrement des sauvegardes, le nom du périphérique, le fuseau horaire configuré, le nom d'hôte et l'OS du dernier équipement sur lequel le terminal a été synchronisé/sauvegardé, etc.

Enfin, la source *MobileInstallation* permet de télécharger le fichier **com.mobile.installation.plist** dans lequel sont stockés les paramètres de lancement de chacune des applications installées sur le terminal.

La bibliothèque **pymobiledevice** mise à disposition sur le GitHub **[IOSFORENSICS]** intègre un exemple d'utilisation de ce service en python : **file_relay.py**.

Service réseau com.apple.pcapd

L'analyse du trafic réseau (hors SMS, MMS et voix) peut être utile lors de l'analyse d'un programme malveillant. L'utilisation du service **com.apple.pcapd** peut répondre à ce besoin sans nécessiter la mise en place d'outillage complexe. En effet, ce service, présent à partir de la version 5 d'iOS, permet la création d'une interface TAP virtuelle sur l'équipement. Le trafic réseau est alors recopié vers le poste d'analyse. Sous Mac OS X, il est possible d'accéder à ce service à l'aide de la commande : **rvictl**.

```
$ rvictl -s <[Device ID]>
Starting device <[Device ID]> [SUCCEEDED]
$ sudo tcpdump -i rvi0 -n
listening on rvi0, link-type RAW (Raw IP) , capture size 65535
bytes
...
```

La bibliothèque **pymobiledevice** **[IOSFORENSICS]** intègre également un exemple d'utilisation de ce service : **pcapd.py**.

2.3.2 Acquisition logique avec altération du système

La technique d'acquisition présentée dans le chapitre précédent n'autorise pas l'accès aux fichiers composant le système. Il est donc nécessaire d'utiliser des techniques de *Jailbreak* afin de déverrouiller le service **com.apple.afc** et d'exposer l'intégralité du système de fichiers. Le système est toutefois altéré, car l'utilisation des techniques de *Jailbreak* implique la modification du système de fichiers. Tous les détails techniques concernant cette technique ont été présentés lors de **[SSTIC 2014]**.

Cette méthode d'acquisition permet au final d'obtenir une **archive du système de fichiers** monté sur un terminal en cours de fonctionnement, ce qui est différent d'une image brute des partitions DATA et SYSTEM obtenue lors d'une acquisition physique.

```
16 ... Oct 13 2013 Applications
16 ... Aug 14 2013 Developer
16 ... Oct 13 2013 Library
16 ... Aug 14 2013 System
16 ... Jun 14 00:26 bin
16 ... Aug 14 2013 cores
16 ... Jun 11 17:49 dev
16 ... 11 Oct 13 2013 etc -> private/etc
16 ... Nov 12 2013 private
16 ... Jun 14 00:18/sbin
16 ... 15 Oct 13 2013 tmp -> private/var/tmp
16 ... Nov 12 2013 usr
16 ... 11 Oct 13 2013 var -> private/var
```

Fichiers accessibles suite à l'acquisition logique.

Les principaux inconvénients d'une telle méthode par rapport à une acquisition physique sont les suivants :



- pas d'accès aux informations HFS+ :
 - ↳ métadonnées incomplètes,
 - ↳ ligne de temps réduite uniquement aux dernières dates de modification ;
- *carving* impossible sur les clusters non alloués.

L'avantage en revanche est que les fichiers du système sont lisibles sans aucune opération de déchiffrement.

3 Analyse

L'analyse des fichiers présents sur le système permet d'identifier la présence de contenu malveillant (traces de *Jailbreak*, documents ou exécutables exploitant une vulnérabilité d'un logiciel tiers, etc.).

3.1 Analyse comparative

Cette méthode d'analyse s'appuie généralement sur un ou plusieurs référentiels de fichiers connus et apporte un temps de gain considérable en identifiant les fichiers présents par défaut sur le terminal.

Ainsi, les analyses comparatives présentées dans les chapitres suivants respectent les étapes ci-dessous :

- comparaison de la liste des noms de fichiers :
 - ↳ avec celle d'un terminal sain de la même version physique et logicielle,
 - ↳ avec celles publiées dans les documentations publiques telles que celles disponibles sur les sites publics [1] [2] [3] [4] ;
- comparaison automatique ou analyse dite par condensat approximatif (*fuzzy hashing*) du contenu des fichiers. Ce type d'analyse permet d'identifier le taux de similitude entre deux fichiers et permet d'identifier les fichiers créés ou modifiés. Un outil tel que **[BINWALLY]** est assez performant pour réaliser ce type d'opération ;
- analyse manuelle des fichiers ajoutés ou altérés, en analysant en priorité ceux créés ou modifiés durant la période de potentielle compromission communiquée par la victime.

3.2 Analyse bas niveau - HFS+

3.2.1 Constitution d'une ligne de temps

Cette opération permet d'obtenir une image de l'activité du système à un instant précis et à travers le système de fichiers HFS+.

Le fichier **\$catalogFile**

Le fichier spécial **\$catalogFile** du système de fichiers HFS+ (utilisé pour les partitions DATA et SYSTEM) contient les métadonnées de chaque fichier et répertoire du terminal (date de création, de dernier accès, de dernière modification, de modification des métadonnées, de sauvegarde), sous la forme de structures **HFSPlusCatalogFile/HFSPlusCatalogFolder** [**\$CATALOGFILE**].

L'emplacement et la taille du **\$catalogFile** sont facilement identifiables puisqu'ils sont indiqués dans l'entête du volume HFS+ (1024 octets à partir de l'offset **0x400**), au sein d'une structure **HFSPlusDataFork** (généralement à l'offset **0x110** de l'entête). Son extraction est donc triviale.

Deux types d'image HFS+

À partir des images physiques HFS+ des partitions DATA/SYSTEM obtenues par une acquisition physique, l'entête du volume HFS+ et le **\$catalogFile** sont facilement identifiables et ne posent pas de difficulté.

Sur un terminal démarré (non *jailbreaké*), une image disque HFS+ au format « *Sparsebundle* » peut être extraite suite à l'interrogation du service **com.apple.mobile.file_relay** (et la source de données *HFSmeta* présentée dans le chapitre 2.3.1). Celle-ci contient l'arborescence du système de fichiers monté mais sans le contenu. Le format de l'image extraite (HFS+) permet par contre de stocker le **\$catalogFile** qui peut ainsi être extrait.

Génération d'une ligne de temps

En plus de supporter l'identification et l'extraction du **\$catalogFile** (*fls*, *icat*), la suite d'outils **[SLEUTHKIT]** propose des options pour générer automatiquement une ligne de temps au format *mactime* depuis une image HFS+ (les deux types d'image étant supportés) :

```
$ fls -m . -r system_201406.dmg > fls_out
```

Mactime (*Sleuthkit*) permet alors de formater les dates et trier les enregistrements :

```
$ mactime -b fls_out -d -z UTC > mactime.csv
```

3.2.2 Identification des fichiers effacés, de leur activité et restauration

Les modifications récentes effectuées sur un fichier peuvent être identifiées à travers le fichier spécial **\$journalFile** (qui utilise partiellement la structure **HFSPlusCatalogFile** et dont l'emplacement est indiqué à l'offset **0xC** de l'entête du volume). En effet, celui-ci contient les données nécessaires au système pour qu'il soit en mesure de restaurer son état suite à un arrêt soudain. Par exemple, si un fichier est créé, modifié, accédé et enfin supprimé, chacune de ces actions entraînera l'ajout d'une nouvelle entrée dans **\$journalFile**. Il y aura donc encore des traces de la

présence du fichier dans **\$journalFile** contrairement à **\$catalogFile** qui efface ses entrées lorsque les fichiers sont supprimés. Toutefois, toutes les transactions ne sont pas enregistrées, **\$journalFile** étant limité en taille (8Mo). L'article **[HFSJOURNALPARSER]** propose une méthode d'analyse pertinente de ces deux fichiers.

En plus d'identifier les fichiers effacés, il est pertinent d'analyser le fichier **\$journalFile** pour identifier les derniers fichiers modifiés. Un programme malveillant, s'il manipule régulièrement des fichiers ou programmes, impactera les entrées de **\$journalFile** et pourra donc être rapidement détecté par ce biais.

En matière d'outillage, l'outil gratuit **[AHJP]** (*Advanced HFS+ Journal Parser*) se basant sur les deux fichiers spéciaux **\$catalogFile** et **\$journalFile** est assez efficace.

Cette opération nécessite toutefois un accès root sur le terminal ou l'obtention d'une image physique.

En ce qui concerne la restauration des fichiers effacés, *Sluehtkit* ne supportant pas l'attribut HFS+ *cprotect* relatif aux classes de protection de chacun des fichiers du terminal, il est nécessaire d'utiliser une alternative. Les chercheurs de « Sogeti ESEC » ont également travaillé sur ce sujet complexe et proposent un script, **emf_undelete.py**, permettant le carving de la mémoire NAND **[LOWLEVEL_IOS]** dans le but d'identifier les fichiers effacés et de tenter leur restauration.

3.3 Analyse de l'arborescence des fichiers

Cette opération peut être réalisée aussi bien à partir d'une image physique que d'une image logique (archive du système de fichiers).

3.3.1 Constitution d'une ligne de temps

Lorsque la date de dernière modification a été conservée pour chacun des fichiers, il est possible de générer une ligne de temps. La commande **tree** permet par exemple d'effectuer cette opération en proposant un format de sortie aisément exploitable pour être trié par date :

```
$ tree --timefmt %Y/%m/%d %H:%M -f -X Applications/
...
<directory name="Applications">
  <directory name="Applications/Test.app" time="2013/10/13 07:33">
    <file name="Applications/Test.app/ATest" time="2013/09/19 07:07"></file>
    <file name="Applications/Test.app/Test.plist" time="2013/10/13 07:33"></file>
    <file name="Applications/Test.app/Default-568h@2x.png" time="2013/09/19 07:07"></file>
    <file name="Applications/Test.app/Info.plist" time="2013/10/13 07:33"></file>
  </directory>
...
```

Affichage des dates de modification du répertoire /Applications.

3.3.2 Analyse différentielle du système de fichiers racine

Il s'agit ici de vérifier si les fichiers constituant le système racine sont intègres et n'ont pas subi de modification malveillante révélatrice d'une compromission. Pour cela, il est nécessaire d'obtenir une image de référence à comparer avec l'image du terminal à analyser. Il existe deux méthodes pour obtenir cette image :

- Depuis une image Firmware IPSW : celle-ci contient les briques de démarrage du système (*iBoot*, *LLB*, *iBEC*, *KernelCache*, ...) ainsi que le système de fichiers racine (*Root File System*) HFS+. Ce dernier, stocké sous la forme d'une image DMG, contient donc les fichiers installés initialement sur le terminal qui peuvent donc être utilisés comme références. L'image DMG est chiffrée par une clé de chiffrement dérivée de la clé GID (propre à chaque famille d'équipement). La clé GID étant uniquement accessible par le processeur cryptographique lors du démarrage du système, il est nécessaire d'avoir recours à l'exploitation d'une vulnérabilité dans la chaîne de démarrage pour en assurer l'extraction. Pour chaque terminal affecté par ce type de vulnérabilité, la communauté *Jailbreak* a mis en place sur l'**[IPHONELINK]** une page centralisant les clés de déchiffrement. L'outil **[VFDECRYPT]** permet notamment le déchiffrement de l'image DMG à partir de la clé de déchiffrement pour permettre par la suite son montage et donc son exploration :

```
$ vfdecrypt -i058-2399-001.dmg -k<key> -odecrypted.dmg
$ dmg2img decrypted.dmg decrypted.img
$ hdiutil attach -readonly -imagekey diskimage-
class=CRawDiskImage \ decrypted.img
...
/dev/disk4s3 Apple_HFSX /Volumes/BrightonMaps10B329.N94
```

- En activant temporairement une copie du service **com.apple.afc** : celle-ci, non soumise aux restrictions imposées par le système, permet d'acquérir l'intégralité des données stockées sur la mémoire flash d'un terminal de référence. Cette méthode d'acquisition, qui utilise les techniques de *Jailbreak* **[POSIXPWN]** **[EVASION]**, a fait l'objet d'un article complet au **[SSTIC 2014]**. Cette technique pourrait être adaptée aux versions 7.1.1 et 7.1.2 en réalisant l'étude du dernier *Jailbreak* en date **[PANGU]**.

Ces données de référence permettent alors d'identifier la modification partielle ou complète des fichiers via une analyse comparative (comparaison des noms de fichiers, du contenu, *fuzzy hashing*, etc.).

3.3.3 Cache noyau (kernelcache)

Sur un système iOS, l'image du kernel (*kernelcache*) se trouve à l'emplacement : **/System/Library/Caches/com.apple.kernelcaches/kernelcache**. Ce fichier



contient le noyau XNU et ses différents modules et extensions (format **.kext**). L'image du noyau est également incluse dans chacune des images du firmware iOS (fichiers IPSW), sous la forme d'un fichier au format **.img3** chiffré. De par la nature sensible de cet élément, il est essentiel de vérifier son intégrité par exemple en le comparant à une image de référence.

Comme pour le système de fichiers racine (voir 3.3.2), le *kernelcache* d'origine est disponible, pour les versions des terminaux vulnérables, sur l'[IPHONEDIKI], accompagné de ses clés de déchiffrement. Il peut donc être utilisé comme référence. Le déchiffrement du *kernelcache*, suivi de la décompression peuvent être réalisés automatiquement via [KCACHE] :

```
./kcache -i kernelcache -v <IV> -k <KEY>
```

La suite d'outils [XPWNT00L] et [LZSSDEC] peut également être utilisée pour réaliser cette opération.

Les deux *kernelcache* obtenus au format exécutable Mach-O (ARM) peuvent alors être comparés aisément via leur condensat (ex. : SHA256). Si ces derniers s'avèrent être différents, une investigation plus poussée doit être menée pour identifier les modifications apportées. La taille du *kernelcache* étant d'une dizaine de Mo, l'analyser avec un désassembleur est fastidieux et complexe. Il est donc préférable, dans un premier temps, de comparer les codes avec des outils plus adaptés tels que [BINDIFF] (comparaison du code désassemblé) et [JOKER] (extraction complète de la table des appels système, de l'emplacement et du nom des extensions noyau).

3.4 Analyse des programmes persistants

3.4.1 Services ou agents lancés au démarrage

L'analyse des services et des agents permet d'identifier les programmes lancés en tâche de fond au démarrage, avec les droits utilisateur ou root. Leur configuration, au format **.plist**, est notamment stockée dans les répertoires suivants :

- agents : **/Library/LaunchAgents** ;
- services : **/System/Library/LaunchDaemons**.

Le contenu des fichiers **.plist** permet d'identifier l'exécutable principal ainsi que les paramètres de lancement. Généralement, l'analyse comparative (voir 3.1) permet de restreindre efficacement l'analyse manuelle à quelques fichiers. Il s'agit ensuite d'effectuer la levée de doute en analysant directement le binaire (voir chapitre suivant pour l'analyse des Mach-O).

3.4.2 Extensions noyau

L'analyse des extensions noyau installées (équivalent aux drivers), autres que celles présentes par défaut sur un système (à travers le *kernelcache*), permet d'identifier les programmes spécifiques pouvant être lancés au niveau de la couche noyau et au démarrage du terminal. Elles sont notamment installées à deux emplacements :

- Extensions System : **/System/Library/Extensions/** ;
- Extensions Library : **/Library/Extensions/**.

Ces dossiers peuvent contenir directement les exécutables, des fichiers **Info.plist**, des plug-ins, etc.

Encore une fois, l'analyse comparative (voir 3.1) permet d'identifier si des extensions autres que celles par défaut ont été installées.

3.4.3 Autres moyens de persistance

D'autres moyens sont couramment employés pour permettre la persistance d'un programme au démarrage notamment :

- **Overrides.plist** : fichier de configuration utilisé par le gestionnaire de services « launchd » pour charger/décharger des services ou agents via l'application launchctl. Ils sont stockés dans plusieurs dossiers selon les droits d'exécution (**/private/var/db/launchd.db/com.apple.launchd*/overrides.plist**) ;
- XPC Services : services utilisés pour la communication entre processus (généralement pour l'authentification partagée) et identifiables dans des dossiers **XPCServices** de **/System/Library/** ;
- **Launchd.conf** (situé dans **/private/etc/**) : fichier permettant l'exécution de commandes à travers *Launchctl* (jusqu'à iOS 6 inclus). Le *Jailbreak* « Evasi0n » utilise cette technique de persistance [ACCUVANT] ;
- **com.apple.mobile.installation.plist** (situé dans **/private/var/mobile/Library/Caches**) : fichier permettant d'identifier les différentes variables d'environnement exportées lors du lancement des applications installées sur le système. Ce fichier étant modifié par diverses versions de Jailbreak, il est important d'y consacrer une partie du temps alloué à l'analyse ;
- **Info.plist** : fichier de configuration de chaque application indiquant le chemin de l'exécutable et qui, modifié, peut permettre le lancement d'une application tierce à la place de celle par défaut.

3.5 Analyse des applications

3.5.1 Identification des applications installées

Comme précédemment évoqué, le service **com.apple.mobile.installation_proxy** (exploitable via l'outil *ideviceinstaller* par exemple) permet d'identifier les applications installées sur l'équipement.

Si le système de fichiers a pu être extrait lors de l'acquisition, il est préférable d'identifier manuellement (donc sans passer par les API d'Apple) les applications installées. Les répertoires courants utilisés pour l'installation des applications d'un terminal iOS sont les suivants :

- **/Applications/MonAppli.app** : répertoire des applications installées par défaut par Apple (non chiffrées) ;
- **/private/var/mobile/Applications/UID/MonAppli.app** : répertoire contenant les applications installées. Ce répertoire contient aussi bien les applications provenant de l'Apple Store dont les exécutables sont protégés par le DRM Fairplay (applications chiffrées) que les applications fournies par un développeur ou une entreprise qui n'embarquent aucun DRM Apple ;
- **/Developer/Applications/MonAppli.app** : répertoire contenant les outils de développement installés à l'aide de l'outil *Xcode*.

L'analyse de chaque application n'est généralement pas possible en termes de temps et le plus simple est d'exclure la liste des applications légitimes des applications potentiellement malveillantes. L'analyse comparative dans ce cas est très efficace et notamment grâce à l'analyse par condensat approximatif. Pour filtrer la recherche, il est également possible d'exploiter la date de création des applications pour concentrer la recherche sur une période spécifique.

3.5.2 Identification des exécutables

Les exécutables Mach-O des applications sont généralement présents dans les répertoires des applications installées (**<rep>/MonAppli.app/MonAppli**).

Un code malveillant n'étant pas forcément rattaché à une application, il est important de lancer une recherche exhaustive sur l'ensemble du système de fichiers et pas uniquement dans les répertoires consacrés aux applications. Le plus simple pour les identifier est de vérifier les entêtes de chaque fichier via par exemple la commande **file**.

```
$ file WWDC.app/WWDC
WWDC.app/WWDC: Mach-O fat file with 2 architectures
```

Encore une fois, l'analyse comparative (voir 3.1) permet de rapidement faire le tri entre les exécutables sains et ceux à analyser.

3.5.3 Analyse des exécutables

Si le système de fichiers a pu être extrait lors de l'acquisition, l'analyse des binaires peut être entreprise.

Il s'agit donc de procéder à l'analyse par rétro-ingénierie des exécutables inconnus qui semblent présenter une menace pour le terminal. *IDA Pro* ou *Hopper Disassembler* peuvent être utilisés.

Le *MISC hors série n°7* consacré au reverse-engineering propose un article très complet sur le reverse iOS et notamment sur l'analyse des briques de sécurité.

Contrairement aux applications système, développeur ou entreprise, les applications provenant de l'Apple Store sont protégées à l'aide du DRM Fairplay. Leur contrôle nécessite un accès root sur le terminal (*jailbreaké*).

Il serait intéressant d'approfondir davantage ce sujet, mais le format Mach-O vaut un article à lui seul. Il est à noter que l'un des articles du *MISC n°67* traite du format Mach-O et de son exploitation.

3.5.4 Analyse des .plist

Il s'agit de vérifier que les fichiers de configuration **.plist** des applications installées n'ont pas été altérés avant l'installation. Une personne malveillante pourrait en effet modifier, dans certains cas, un fichier de configuration, la vérification de la signature réalisée par Apple ne se basant que sur l'exécutable et non sur le fichier de configuration. Les fichiers de configuration permettant l'exécution de l'application sont stockés dans le répertoire racine de l'application sous le nom **Info.plist** :

```
$ plutil -convert xml1 WWDC.app/Info.plist
$ less WWDC.app/Info.plist
...
    <string>ShareUnlock</string>
    <key>CFBundleExecutable</key>
    <string>../../../../../../../../var/mobile/Media/Downloads/WWDC.
app/WWDC</string>
...
```

Exemple de Info.plist modifié permettant le lancement d'une application malveillante.

D'autres fichiers **.plist** peuvent aussi être présents dans le répertoire de l'application afin de stocker des paramètres de configuration.



L'ouverture des fichiers **.plist** au format XML peut s'effectuer dans un éditeur de texte alors que l'ouverture du format binaire nécessite un outillage spécifique pour la lecture ou la conversion (*Xcode* sous Mac, *libplist* sous Linux [**LIBIMOBILEDEVICE**] ou [**PLIST_EDITOR**] sous Windows).

3.5.5 Signature des Mach-O et autorités de certification installées

Vérification des AC supplémentaires installées

Un terminal iOS laisse la possibilité à l'utilisateur d'installer son propre certificat généré par Apple lorsque celui-ci possède un compte entreprise ou développeur. Dans ce cas, si le certificat est installé sur le terminal, les applications signées avec la clef privée de ce certificat peuvent être installées sans passer par le processus de vérification d'Apple. Ainsi, une application malveillante pourrait être installée sur l'équipement par toute personne ayant dérobé un certificat entreprise ou développement. Comme précédemment évoqué, le service **com.apple.misagent** (via l'outil **ideviceprovision** par exemple) permet d'extraire les certificats des autorités de certification installés, sur un terminal *jailbreaké* ou non.

Signature des Mach-O

En plus des autorités de certification installées, il peut être intéressant de vérifier si tous les binaires utilisés par le système sont signés par une autorité de confiance. L'outil **codesign** permet de simplifier cette opération en réalisant l'extraction des informations concernant la signature. Par exemple, un Mach-O signé par Apple directement (*iPhone OS Application Signing*) pourra être installé sur tous les appareils :

```
$ codesign -dvvvv Le\ Monde.fr
Executable=/Volumes/iOS/private/var/mobile/Applications/C24AC...
Authority=Apple iPhone OS Application Signing
Authority=Apple iPhone Certification Authority
Authority=Apple Root CA
...
```

En revanche, une application signée par un certificat développeur ou entreprise ne pourra être installée que sur un appareil (celui du développeur) ou après installation du certificat de l'entreprise :

```
$ codesign -dvvv PPHelperNS
Executable=PPHelperNS.app/PPHelperNS
...
Authority=iPhone Developer: caijuan huang (89S3J5LGSWL)
Authority=Apple Worldwide Developer Relations Certification
Authority
Authority=Apple Root CA
...
```

Évidemment afficher le CN (*Common Name*) du certificat ne permet pas de s'assurer qu'il s'agisse d'un certificat de confiance. L'extraction de tous les

certificats stockés dans l'exécutable permet alors de s'en assurer :

```
$ codesign -dvvvv -extract-certificates PPHelperNS
```

Les certificats sont alors stockés dans les fichiers « *codesign0*, *codesign1*, ... » au format DER (*codesign0* étant celui ayant signé l'exécutable, les autres étant des AC intermédiaires et root). Il est alors possible de les transformer au format PEM via **openssl** :

```
$ openssl x509 -inform DER -in codesign0>certificat0.pem
```

Les certificats des différentes autorités de certification doivent ensuite être concaténés au sein d'un même certificat :

```
$ cat certificat1.pem >> alltrusted.crt
$ cat certificat2.pem >> alltrusted.crt
...
```

La validité du certificat et de sa chaîne de certification peut alors être confirmée via l'option « **verify** » d'**openssl** et les certificats des AC (ajout de **-ignore_critical** parfois nécessaire) :

```
$ openssl verify -CAfile alltrusted.crt certificat0.pem
certificat0.pem: OK
```

La moindre erreur concernant le certificat pourra donc être détectée par ce biais (date de validité, chaîne de confiance, certificat auto-signé, absence de certificat, etc.) [**OPENSSL_VERIFY**].

Cependant, la validité du certificat de l'autorité de certification racine d'Apple inclus dans l'exécutable, soit *Apple Root CA*, doit aussi être vérifiée puisque les précédentes opérations se basent en partie sur celui-ci. Le certificat officiel d'Apple étant disponible sur son site [**APPLE_ROOT_DER**], l'opération consiste à comparer celui-ci avec celui extrait de l'exécutable (généralement *codesign2*).

3.5.6 Analyse dynamique des exécutables

En investigation numérique mobile, l'analyse dynamique ne sera généralement réalisée qu'en dernier recours.

En effet, l'analyse dynamique ne peut être réalisée que dans les deux scénarios suivants :

- l'équipement à analyser est *jailbreaké* et permet alors l'installation d'outils de monitoring des applications ;
- l'application « suspecte » a été identifiée et son binaire peut être à nouveau installé sur un terminal d'analyse ou un émulateur. Le *Jailbreak* de l'équipement n'est alors pas nécessaire.

L'un des articles du MISC Hors-série n°7 consacré à iOS propose déjà les moyens de débogage possible pour analyser une application en espace utilisateur (*gdb*, *lldb*, interposition...). D'autres moyens non évoqués dans l'article permettent l'analyse dynamique :

- utilisation de la bibliothèque **[MOBILESUBSTRATE]** ;
- utilisation de **[CYCRYPT]**.

Certaines de ces méthodes ont notamment été présentées à **[GREHACK_2012]** et **[HACK.LU_2012]**.

Il serait intéressant d'approfondir davantage ce sujet, mais l'analyse dynamique vaut un article à elle seule.

3.6 Activité système et réseau en temps réel

3.6.1 Exploitation des journaux Syslog

Comme présenté au début de l'article, le service **com.apple.syslog_relay** permet d'extraire les journaux *Syslog* générés par les processus (outil **idevicesyslog**) :

```
ASL is here to serve you
>
Jul 23 23:42:06 demo installld[53] <Notice>:
0x241000 handle_install_for_ls: Install of "/var/mobile/Media//
Downloads/pkg.zip" requested by mobile_installation_proxy
Jul 23 23:42:07 demo installld[53] <Error>:
0x241000 stage_package: Could not peruse package at /var/tmp/
install_staging.3E1vf0/fo0_extracted
Jul 23 23:42:07 demo mobile_installation_proxy[357] <Notice>:
001c1000 _send_message: Could not send message: Broken pipe
Jul 23 23:42:07 demo mobile_installation_proxy[357] <Error>:
0x1c1000 installation_callback: Could not send response to host
Jul 23 23:42:07 demo mobile_installation_proxy[357] <Warning>:
ERROR: MobileInstallationInstallForLaunchServices returned nil
Jul 23 23:42:07 demo mobile_installation_proxy[357] <Error>:
0x1c1000 handle_install: Installation failed: Error
Domain=LaunchServicesError Code=0 "The operation couldn't be
completed. (LaunchServicesError error 0.)" UserInfo=0x1755a810
{Error=PackageInspectionFailed}
```

Traces générées lors d'une exploitation d'une vulnérabilité (Evasi0n7).

3.6.2 Flux de communication

Comme précédemment évoqué, l'activité réseau transitant par les différents canaux radio (3G/4G, Wi-Fi) peut être répliquée sur une interface réseau virtuelle à travers la connectique USB du terminal (utilisation du service **com.apple.pcapd**). Cette technique permet d'une part d'identifier les processus générant du trafic réseau et d'autre part d'extraire le contenu des flux réseau au format standard **.pcap** (exploitable donc

via *Wireshark*). La connexion du terminal vers un serveur C&C peut être rapidement identifiée à l'aide de cette technique.

Les flux SSL ne sont pas déchiffrables par ce biais. Afin de pouvoir analyser ces flux, il est nécessaire de réaliser une attaque de type « homme du milieu » (MITM). La machine de l'analyste doit donc être en coupure entre la sortie Internet et le terminal par le biais d'un réseau Wi-Fi. Il est nécessaire d'installer sur le terminal le certificat de l'autorité de certification ayant signé le certificat serveur afin que :

- le certificat serveur présenté par la machine de l'analyste au terminal soit fiable (car signé par une autorité de confiance) ;
- l'analyste puisse déchiffrer le contenu capturé.

Cependant, dans le cas où l'application vérifie si le certificat du serveur (présenté alors par la machine de l'analyste) est bien celui attendu, il est nécessaire de désactiver ce mécanisme de sécurité, plus connu sous le nom de « *certificate pinning* », pour que l'application initie des connexions SSL avec la machine de l'analyste. Les outils **[IOS_KILL_SWITCH]** **[SSL_DISABLE]** permettent de réaliser cette opération en modifiant les fonctions responsables du contrôle SSL sur un terminal *jailbreaké*.

Par ailleurs, la présence de traces indiquant l'utilisation de ces méthodes de capture est généralement signe qu'une interception des flux du terminal a été tentée. L'outil **keychain_tool** **[DATA-PROTECTION]** permet entre autres d'extraire les certificats SSL installés sur le terminal.

```
$ python python_scripts/keychain_tool.py -dsc keychain-2.db
a6eec3ba124cb7fa.plist
...
16 |PortSwigger CA |com.apple.certificates|AlwaysThisDeviceOnly|
```

Extraction d'un certificat de capture SSL (Burp).

3.7 Journaux d'historique

3.7.1 Fichiers de journalisation

Il s'agit d'analyser, dans les fichiers de journalisation, toutes les traces laissées par le système, les services et les applications lors de leur exécution ou leur crash, dans le but de tenter l'identification des indices de compromission. Comme précédemment évoqué, le service **com.apple.crashreportmover** (via l'outil **idevicecrashreport**) permet d'extraire les rapports générés par le système lors du crash d'une application ou d'un processus et pouvant ainsi parfois révéler la présence d'un programme malveillant instable ou modifiant des fonctions système importantes.



Si le système de fichiers a pu être extrait lors de l'acquisition, tous les journaux de l'équipement, stockés à différents emplacements, peuvent être analysés pour rechercher des indices de compromission :

- `/private/var/log/` ;
- `/private/var/logs/` ;
- `/private/var/mobile/Library/Logs/` ;
- `/private/etc/asl/`.

Cependant, pour éviter de passer à côté d'un fichier important ou inconnu et généré par une application tierce, il est conseillé d'effectuer une recherche manuelle des fichiers `*.log`, `*.asl` et des répertoires `*log*` et `*asl*`.

```
$ find /Volume/Data -iname "*.log"
./MobileSoftwareUpdate/restore.log
./logs/AppleSupport/general.log
./logs/keybagd.log
...
./mobile/Media/redsn0w_logs/redsn0w_20140623015110.log
```

Comme l'illustre l'exemple ci-dessus, le nom de certains fichiers peut révéler la présence de *Jailbreak* du terminal.

Alors que les fichiers `.log` sont lisibles directement avec un éditeur de texte classique, les fichiers binaires au format Syslog `.asl` nécessitent d'être décodés avec la commande `Syslog` sous Mac OS X ou avec un outil spécifique tel que `ccl_ccldb.py` [`CCL_ASL`].

■ ASTUCE

Les fichiers de logs ASL de diagnostic stockés dans le répertoire `/private/var/log/DiagnosticMessages/` stockent des informations concernant le chargement et déchargement de la batterie. Il est ainsi possible de savoir, lorsque la charge augmente, les dates précises de chargement et donc potentiellement de branchement USB « malveillant » sur le poste d'un attaquant. Le script Python `iOS_asl_power_timeline.py` [`CCL_ASL`] permet d'exploiter automatiquement ces fichiers :

```
25/06/2014 13:34:00,82,BATT
25/06/2014 13:45:03,80,BATT
25/06/2014 13:45:09,80,BATT
25/06/2014 13:45:32,80,BATT
25/06/2014 13:45:43,80,BATT
30/06/2014 09:39:23,97,BATT
```

Statistiques sur la consommation de la batterie.

3.7.2 Historique de navigation Web

Il s'agit d'identifier si un éventuel programme malveillant ou un certificat d'autorité malveillante a pu être téléchargé sur le terminal par le biais du

navigateur Web Safari. Les fichiers d'historique de navigation de Safari sont stockés dans deux bases de données SQLite :

- `/private/var/mobile/Library/Caches/com.apple.WebAppCache/ApplicationCache.db` ;
- `/private/var/mobile/Applications/7573AC5A-A303-4C4A-B182-85C68BC14BDF/Library/Caches/com.apple.mobilesafari/Cache.db`.

Pour rappel, la méthodologie de *Jailbreak* « *jaibreakme.com* » permettait, suite au téléchargement sur Internet et à l'ouverture d'un document PDF, de débriquer le terminal en exploitant une vulnérabilité dans le lecteur de fichier PDF.

Conclusion

Cette introduction à l'investigation numérique sur terminal iOS montre clairement que l'acquisition de données reste possible sur ce type d'équipement, malgré les accès limités et les mécanismes de protection mis en œuvre. L'analyse des données collectées, quant à elle, est proche en matière de méthodologie d'une analyse d'un ordinateur et quasi similaire à celle menée sur un environnement Mac OS X. De nombreuses recherches ayant été menées sur le sujet, il existe un nombre important d'outils permettant une collecte et une analyse fiable. Enfin, nous retiendrons que les moyens d'acquisition présentés sont inefficaces sur un terminal récent verrouillé, cas peu courant dans le cadre légitime de recherche d'indices de compromission. ■

■ REMERCIEMENTS

À titre personnel, nous tenons à remercier Jean Sigwald ainsi que les autres développeurs pour leurs outils.

■ RÉFÉRENCES

[SSTIC 2012] https://www.sstic.org/media/SSTIC2012/SSTIC-actes/forensicsios/SSTIC2012-Slides-forensicsios-sigwald_bedrone.pdf

[SSTIC 2014] https://www.sstic.org/media/SSTIC2014/SSTIC-actes/Mathieu_RENARD_-_Investigation_numerique_iOS/SSTIC2014-Article-Mathieu_RENARD_-_Investigation_numerique_iOS-renard.pdf

[DATA-PROTECTION] <https://code.google.com/p/iphone-dataprotection/>

[LIMERAİN] <http://www.limera1n.fr/>

[UFED] <http://www.cellebrite.com/fr/mobile-forensics/products/standalone/ufed-touch-ultimate>



■ SUITE DES RÉFÉRENCES

[LIBIMOBILEDEVICE] <http://www.libimobiledevice.org/>
[DEVICEINFO] <https://github.com/bryanforbes/libimobiledevice/blob/master/tools/deviceinfo.c>
[DEVICEDATE] <https://github.com/libimobiledevice/libimobiledevice/blob/master/tools/devicedate.c>
[DEVICEINSTALLER] <https://github.com/libimobiledevice/deviceinstaller>
[IFUSE] <https://github.com/libimobiledevice/ifuse>
[DEVICEPROVISION] <https://github.com/libimobiledevice/libimobiledevice/blob/master/tools/deviceprovision.c>
[DEVICESYSLOG] <https://github.com/libimobiledevice/libimobiledevice/blob/master/tools/devicesyslog.c>
[DEVICECRASHREPORT] <https://github.com/libimobiledevice/libimobiledevice/blob/master/tools/devicecrashreport.c>
[DEVICEBACKUP] <https://github.com/mcoyer/libiphone/blob/master/tools/devicebackup.c>
[DEVICEBACKUP2] <https://github.com/libimobiledevice/libimobiledevice/blob/master/tools/devicebackup2.c>
[IOSFORENSICS] <https://github.com/iOSForensics>
[1] <https://github.com/limneos/classdump-dyld/tree/master/iphoneheaders/iOS7.0.3/Applications>
[2] <https://github.com/Legoless/iOS-Localization/tree/master/7.0.4/iPhone/Applications>
[3] <https://github.com/MP0w/iOS-Headers/tree/master/iOS7>
[4] <http://vladalexa.com/iosx/sources/>
[BINWALLY] <https://github.com/bmaia/binwally>
[\$CATALOGFILE] <https://developer.apple.com/legacy/library/technotes/tn/tn1150.html#CatalogFile>
[SLEUTHKIT] <http://www.sleuthkit.org/>
[HFSJOURNALPARSER] http://www.kazamiya.net/en/HFSJournalParser_TrackFileActivity
[AHJP] https://docs.google.com/forms/d/1_Zrf7LfmnkIJfJ7CtecdAiAWGdRkNp2ltqqHuYFncQ/viewform
[LOWLEVEL_IOS] <http://esec-lab.sogeti.com/post/Low-level-iOS-forensics>
[IPHONEWIKI] http://theiphonewiki.com/wiki/Firmware_Keys
[VFDECRYPT] <http://code.google.com/p/iphone-elite/downloads/list>
[P0SIXPWN] <http://p0sixspwn.com/>
[EVASION] <http://evasi0n.com/>
[PANGU] <http://en.pangu.io/>
[KCACHE] <https://github.com/badeip/kcache>
[XPWNTOOL] <http://theiphonewiki.com/wiki/Xpwntool>
[LZSSDEC] <http://nah6.com/~itsme/cvs-xddevtools/iphone/tools/lzssdec.cpp>
[BINDIFF] <http://www.zynamics.com/bindiff.html>
[JOKER] <http://newosxbook.com/src.jl?tree=listings&file=8-joker.c>
[ACCUVANT] <http://blog.accuvant.com/bthomasaccuvant/evasi0n-jailbreaks-userland-component/>
[PLIST_EDITOR] <http://www.icopybot.com/plist-editor.htm>
[APPLE_ROOT_DER] <https://www.apple.com/appleca/AppleIncRootCertificate.cer>
[OPENSSL_VERIFY] <https://www.openssl.org/docs/apps/verify.html>
[MOBILESUBSTRATE] <http://www.cydiasubstrate.com/>
[CYCRYPT] <http://www.cycrypt.org/>
[GREHACK_2012] <http://goo.gl/wfkXpL>
[HACK.LU_2012] <http://goo.gl/1UhtIZ>
[IOS_KILL_SWITCH] <https://github.com/iSECPartners/ios-ssl-kill-switch>
[SSL_DISABLE] https://github.com/pod2g/ios_stuff/blob/master/misc-hs07/ssl_disable.c
[CCL_ASX] <https://code.google.com/p/ccl-asx/>

QUELLE MÉTHODOLOGIE POUR UNE RÉPONSE À INCIDENT SUR ANDROID ?

Lexfo - contact@lexfo.fr



mots-clés : ANDROID / RÉPONSE À INCIDENTS / BOOTLOADER / ADB / MÉMOIRE VIVE / MDM

Les terminaux mobiles sous Android sont aujourd'hui autant ciblés par les attaquants que les postes de travail ou serveurs pour tenter de s'introduire sur les systèmes d'information des entreprises. Ces dernières doivent donc s'organiser pour prendre en compte dans leur processus de réponse à incident ces nouveaux vecteurs d'attaque. Quelle est la complexité d'un tel sujet ? Les outils existants suffisent-ils ? Comment agir sur une flotte d'entreprise ? Ce sont autant de questions auxquelles nous allons essayer de répondre.

1 Introduction

La hausse du marché du smartphone impacte bien évidemment les entreprises : les collaborateurs souhaitent pouvoir consulter leurs mails, leurs agendas, accéder à l'intranet de l'entreprise, etc. quel que soit le lieu géographique où ils se trouvent. De facto, les terminaux mobiles sont à la fois connectés à Internet et aux systèmes d'information des entreprises.

Partant de ce constat, il est évident que le smartphone (au sens large du terme) devient une cible pour un attaquant souhaitant s'introduire dans le système d'information d'une entreprise. Les processus de réponse à incident, lors d'une compromission de SI, doivent par conséquent prendre en compte ces « nouveaux » équipements. Ils peuvent être différents selon que l'entreprise possède ou non une flotte professionnelle de smartphones et tablettes, et gère cette flotte avec une solution MDM (*Mobile Device Management*). Dans le cas où l'entreprise n'a pas atteint cette maturité, se pose alors le problème de l'hétérogénéité du parc, c'est-à-dire de la version du constructeur, du

système d'exploitation et des applications installées. Ces problématiques sont accentuées lorsqu'il s'agit des terminaux Android.

Même si environ trois quarts des utilisateurs Android possèdent un terminal en version Jelly Bean (4.1.x, 4.2.x, 4.3.x) sortie seulement en juillet 2012, le quart restant peut posséder l'une de ces 4 versions d'OS : *Froyo* (2.2), *Gingerbread* (2.3.3 - 2.3.7), *Ice Cream Sandwich* (4.0.3 - 4.0.4), *KitKat* (4.4). Côté constructeur OEM, le monde Android est tout autant fragmenté. Il semblerait que Samsung détienne environ deux tiers des parts du marché, suivi par les 4 constructeurs que sont Motorola, Sony, HTC, LG qui se partagent globalement le reste du marché (même si nous n'avons pas cité Xiaomi qui est la marque de smartphone la plus vendue en Chine ou encore Wiko, société française).

Cet article tentera donc de décrire dans les grandes lignes, car quelques pages ne suffisent évidemment pas à être exhaustif, comment procéder à une analyse forensics sur système Android suite à une intrusion sur le SI de l'entreprise, que ce soit depuis un terminal isolé ou appartenant à une flotte professionnelle gérée par une solution MDM.



2 Contexte et objectifs

2.1 Contexte

De manière générale, une investigation informatique sur un support numérique doit suivre la logique générale suivante :

- accès au support ;
- acquisition des données ;
- analyse des données.

Sur un terminal Android, cette investigation peut s'avérer complexe puisqu'elle est fortement liée aux constructeurs (ou OEM), aux opérateurs de téléphonie mobile (les OEM et opérateurs peuvent s'associer pour ajouter au téléphone des briques logicielles opérateurs), à la version de l'OS Android et à sa configuration. Cette complexité se retrouve particulièrement au cours de l'étape d'acquisition des données.

Selon le contexte et l'environnement dans lequel l'analyste se retrouve, nous verrons dans la suite de l'article que l'obtention des privilèges *root* sur le téléphone peut s'avérer nécessaire. Un rappel sur la façon de procéder sera donc utile.

2.2 Objectifs

Les objectifs généraux d'une réponse à incident en cas de compromission sur un SI d'une entreprise sont clairs. Elle doit idéalement pouvoir répondre à l'ensemble de ces questions : comment l'attaquant s'est introduit, des portes dérobées ont-elles été déposées au sein du SI, y a-t-il eu ex-filtration d'information et par quels moyens ?

Si nous nous mettons à la place d'un attaquant, la compromission d'un terminal Android va nous permettre certes de récupérer des informations intéressantes sur ce dernier, mais aussi et surtout de rebondir sur le SI de l'entreprise auquel est interconnecté le mobile. L'investigation d'un terminal Android doit donc se concentrer sur ce point : quelles informations présentes sur le téléphone ont pu être utilisées par l'attaquant pour s'introduire sur le SI et sur quel équipement a-t-il rebondi (un portail Web, un extranet, un *store* privé, etc.) ?

Bien qu'il y ait de meilleures solutions, un attaquant peut choisir d'utiliser le mobile Android pour revenir sur le SI compromis ou même pour exfiltrer de l'information. L'attaquant peut aussi vouloir géolocaliser de manière précise le mobile. La recherche d'éventuelles portes dérobées est donc nécessaire, qu'elles soient dans les applications Android de nature *userland*, *kernelland* ou tout en mémoire, cette dernière n'étant malgré tout pas très judicieuse (faible persistance à cause des problèmes de batterie).

Enfin pour terminer, un attaquant peut aussi vouloir cibler une personne plutôt qu'une entreprise. La compromission du téléphone servira donc à récupérer les mails de l'utilisateur, ses appels et SMS. Cet aspect de l'analyse forensics n'est pas abordé dans le cadre de cet article, mais la recherche de portes dérobées peut aussi s'orienter dans ce sens.

3 Accès au terminal : conditions optimales

L'accès au terminal peut se faire de deux manières différentes : logique ou physique. Ce dernier nécessite un minimum de compétences techniques en électronique et dépend fortement du modèle constructeur (accès au port JTAG, dessoudage de la mémoire flash, etc.), il est donc peu réaliste de procéder à une telle démarche dans un cadre professionnel : absence de moyens, de compétences et faible réactivité. L'accès logique est donc privilégié.

Pour faciliter l'accès au terminal, le système Android possède une interface de *debug* sur USB nommée « *Android Debug Bridge* » qui permet plusieurs actions intéressantes dans le cadre d'une investigation informatique, comme la possibilité d'effectuer des sauvegardes, d'accéder à un *shell* ou encore de récupérer des fichiers.

Afin que cette interface soit opérationnelle, le mode *debug* USB du terminal doit être activé. C'est une option souvent non activée par défaut accessible uniquement une fois le terminal déverrouillé.

Par ailleurs, depuis Android 4.2.2, une protection a été implémentée pour éviter l'exécution de commande **adb** lorsque l'hôte n'est pas un hôte de confiance. Il est possible de réaliser cette action que si le terminal n'est plus verrouillé : une demande de confirmation apparaît sur le terminal.

Dans un contexte de réponse à incident, l'activation et l'utilisation d'**adb** sont non bloquantes puisque l'analyste est supposé pouvoir déverrouiller le terminal.

4 Acquisition des données

4.1 Acquisition de la carte SD

Si le terminal contient une carte SD, la première étape est de copier son contenu. L'acquisition (copie bit à bit) d'une carte SD est classique.

L'insertion d'une carte SD vierge dans le terminal permettra ensuite d'avoir un espace de stockage utile pour la suite de l'investigation.

4.2 Acquisition des informations natives

Plusieurs informations disponibles nativement sur le système et ne nécessitant aucun privilège particulier sont utiles à acquérir. Ces informations sont obtenues à l'aide de différentes commandes.

4.2.1 Les commandes classiques Linux

Plusieurs commandes Linux sont disponibles via **adb** sans même avoir besoin d'installer *Busybox* : **ps**, **netstat**, **lsmod**, **dmesg**, etc. Elles permettent d'établir un premier état des lieux :

```
> adb shell netstat
```

4.2.2 Les commandes ADB

getprop

Cette commande permet de lister les propriétés système du terminal :

```
> adb shell getprop
```

logcat

Logcat permet d'afficher les journaux d'événements en temps réel, les fichiers logs en question étant **/dev/log/system** et **/dev/log/main**.

```
> adb logcat : logs de différentes applications et d'une partie du système
> adb logcat -v [time|process] : ajoute les timestamps ou les PID
> adb logcat -b [main|system] : les logs main par défaut ou system
> adb logcat -h : l'aide complète
```

Les logs étant en temps réel, la commande **logcat** n'affiche qu'environ les 3 dernières heures d'activité du système.

dumpsys

Cet autre utilitaire permet d'obtenir davantage de logs systèmes et applicatifs :

```
> adb shell dumpsys : information à propos des services systèmes
> adb shell dumpsys | grep DUMP : permet de lister les modules de dumpsys afin d'obtenir davantage d'informations. Une commande équivalente :
> adb shell service list
```

Parmi les modules intéressants :

```
> adb shell dumpsys account : accounts sur le device
> adb shell dumpsys activity : activité de l'OS (processus, tâches, services, etc.)
> adb shell dumpsys package : informations détaillées sur les packages
```

4.3 Acquisition des données

La suite de l'investigation consiste à acquérir les données présentes sur le système Android, que ce soit les données systèmes, les configurations, les applications et leurs données, etc. À nouveau, selon la version d'Android, le constructeur, etc., différentes méthodologies se présentent.

4.3.1 Android 4 : Android Backup Manager

Depuis Android 4.0, une fonctionnalité permettant une sauvegarde du terminal a été mise en place, accessible par le client **adb** et la commande « **adb backup** ». Selon Google, cette fonctionnalité de sauvegarde est disponible pour près de 85% des terminaux. Sans avoir besoin de droits *root*, et c'est là tout l'intérêt, il est ainsi possible d'obtenir une sauvegarde du téléphone comprenant :

- les APK des applications ;
- les données de chaque application ; - les applications systèmes ;
- les données présentes dans la carte SD.

Cependant, cette option d'**adb** est limitée puisque le contenu complet du terminal n'est pas récupéré. Pour effectuer la sauvegarde, la commande est la suivante :

```
> adb backup -all -apk -shared -system -f backup2.ab
```

Une fois cette commande réalisée, le système demande alors si vous souhaitez bien réaliser une sauvegarde et si vous souhaitez chiffrer celle-ci. Il est donc nécessaire que le terminal ne soit pas verrouillé pour confirmer le lancement de la sauvegarde côté téléphone.

Le format de sauvegarde Android est une archive *tar* compressée avec un entête particulier, il est possible d'obtenir un fichier *tar* classique à l'aide de cette commande :

```
$ dd if=mybackup.ab bs=24 skip=1 | openssl zlib -d > mybackup.tar
```

Des outils existent cependant pour regarder le contenu d'une sauvegarde comme par exemple ABE (*Android Backup Extractor*) [1].

4.3.2 Packages manager

Le système Android, et plus précisément le service *Packages Manager*, enregistre toutes les applications installées ainsi que les permissions correspondantes dans le fichier XML **/data/system/packages.xml**.



Ce dernier n'est en revanche pas accessible si l'on ne détient pas les droits de l'utilisateur system ou supérieur.

Cependant, Android fournit le binaire `/system/bin/pm` permettant d'interroger le service afin d'obtenir un contenu formaté de toutes les informations qu'il contient.

Les commandes suivantes sont notamment intéressantes :

```
> pm list packages : liste des applications installées sur le terminal
> pm list packages -3 : liste des applications third-party
> pm list permissions : liste complète des permissions utilisées par toutes les applications du système
> pm list permissions -d : liste des permissions utilisées par des paquets et considérées comme dangereuses.
> pm path com.android.android : affiche le chemin de l'apk correspondant
> pm -lf : liste l'ensemble des applications avec le chemin de l'apk et le nom de l'application associée
```

4.3.3 Acquisition des privilèges root

Il est possible que l'analyste arrive à une situation où il se trouve limité par les informations obtenues et ait besoin du niveau de privilèges *root* pour aller plus loin.

Le terminal Android peut par exemple être dans une version inférieure à 4.0 et ne dispose donc pas du service de backup permettant l'accès aux données des applications, ou bien il est nécessaire d'analyser le terminal dont le système lui-même a été compromis (*rootkit*) ou encore, il est nécessaire d'obtenir les potentielles données supprimées sur le système de fichiers.

La mise en place d'un accès *root* modifie l'état des données sur le téléphone et détruit certaines informations. Néanmoins un terminal mobile avec un accès *root* nous permet d'obtenir un accès complet, qu'il s'agisse des données utilisateur ou système.

Le bootloader

Le *bootloader* est la toute première chose qui s'exécute sur le terminal. À l'instar d'un BIOS, celui-ci se charge d'initialiser les différents composants matériels. Il charge ensuite le noyau en mémoire qui réalisera toutes les tâches nécessaires à l'exécution de la ROM et passera la main au processus *init*. Le *bootloader* est un programme développé par le fabricant et donc spécifique à chaque constructeur voire à chaque modèle.

Le mode Download

Le *bootloader* peut être également utilisé pour démarrer le terminal dans ce qui est appelé le *Download Mode* permettant alors de flasher l'appareil et modifier le firmware. Cette opération sera alors réalisée avec le protocole *fastboot* ou un autre protocole propriétaire tel que *Odin* en ce qui concerne la gamme Samsung par

exemple ou *SBF* pour Motorola. Le programme *fastboot* permettant d'exploiter ce protocole est disponible dans le SDK Android. L'accès au *Download Mode* se fait généralement par une combinaison de touches particulière lors du *boot*.

Faisons un focus sur les terminaux Samsung puisque ce sont les plus répandus. Si aucun mécanisme de vérification de signature du noyau n'existe dans le *bootloader*, il est alors possible de profiter d'*Odin* [2], le logiciel de « *recovery* » de Samsung qui permet de flasher de nouveaux composants sur le terminal sans vérification préalable.

La mise à jour s'effectue via les outils *Odin v3* ou *Heimdall*. Une mise à jour se présente sous la forme d'un fichier zip contenant un script *shell* se faisant interpréter lors du flashage d'une ROM. Cette dernière représente elle-même la partition `/system` et des parties de la partition `/data` du téléphone. Lors de ce flashage, les partitions sont en lecture/écriture permettant alors de modifier le système interne du système d'exploitation Android. Le terminal est désormais *rooté*.

L'obtention des privilèges *root* sur un terminal Android passe donc par un *reboot* du terminal.

Il est évident que le flashage du terminal ainsi qu'un *reboot* ont une incidence pour une investigation à l'instar d'un *reboot* serveur ou poste de travail.

Le mode Recovery

De la même manière que pour le *Download Mode*, le *bootloader* peut également permettre d'accéder à un mode de récupération. Le mode de récupération d'Android est le mécanisme utilisé pour la mise en place des mises à jour sur le système sans effacer les données utilisateurs. Il consiste en fait en un noyau Linux minimal présent sur la partition `/recovery`.

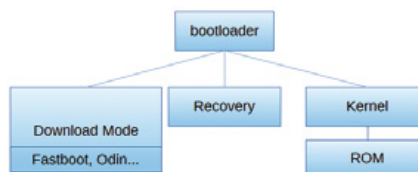
Deux *recovery* alternatifs populaires et permettant de réaliser des sauvegardes d'un terminal sont *TWRP* et *ClockWorkMod*.

Les bootloaders verrouillés

Le *bootloader* est dit *locked* lorsqu'il inclut un mécanisme cryptographique empêchant le démarrage ou la modification du firmware par du code non signé. La plupart des terminaux Android ont un *bootloader* signé. Cependant, certains terminaux bon marché n'ont pas de *bootloader* bloqué et beaucoup de constructeurs permettent le déblocage du *bootloader* à travers le protocole *fastboot* et la commande :

```
> fastboot oem unlock.
```

Dans certains cas, le constructeur propose un portail web permettant le déblocage de l'appareil. C'est le cas des constructeurs HTC, Sony et Motorola.



Dans la mesure où un *bootloader* débloqué permet la modification du *firmware*, il est alors possible de modifier l'image *recovery* ou le *firmware* pour s'octroyer un accès root et avoir un accès intégral aux données. Il est important d'avoir en tête que le modèle de sécurité du terminal repose sur le fait que le *bootloader* réalise cette vérification de l'image sur laquelle il démarre. Dès lors que le *bootloader* est débloqué, un attaquant peut alors obtenir toutes les données du terminal dans la mesure où celles-ci ne sont pas chiffrées. Pour cette raison, le déblocage du *bootloader* provoque une suppression des données utilisateur lorsque celui-ci est réalisé. En conséquence, le déblocage du terminal ne sera pas tenté au cours d'une réponse à incident au risque de perdre certaines informations.

Vulnérabilité noyau

Dans le cas où le *bootloader* n'est pas déverrouillé, il est toujours possible d'obtenir un accès *root* en exploitant une vulnérabilité du noyau (ou d'un démon Android). Des vulnérabilités ont été trouvées et rendues publiques par le passé, notamment dans le démon **adb**, dans le système de *backup*, dans le système de vérification des signatures de mise à jour ou encore dans la gestion des conteneurs sécurisés.

Une des dernières vulnérabilités noyau en date, qui exploite une faille dans l'appel système *futex*, n'a pas encore été *patchée* et rend facilement possible l'obtention du *root* à l'aide d'une application réalisée par George Hotz [3].

4.3.4 Acquisition complète

À partir de l'accès *root*, si l'on souhaite obtenir un *dump* des partitions, il est possible de procéder comme sous Linux.

La commande **mount** indique les points de montage et les *devices* utilisés. La liste des partitions par nom de *device* est obtenue par la commande suivante :

```
> ls -al /dev/block/platform/msm_sdcc.1/by-name/
lrwxrwxrwx root root 1971-04-26 20:45 aboot -> /dev/block/mmcblk0p6
lrwxrwxrwx root root 1971-04-26 20:45 apnhtos -> /dev/block/mmcblk0p1
lrwxrwxrwx root root 1971-04-26 20:45 backup -> /dev/block/mmcblk0p23
lrwxrwxrwx root root 1971-04-26 20:45 boot -> /dev/block/mmcblk0p20
[...]
```

Et le *dump* d'une partition s'effectue bien évidemment avec la commande « **dd** » (en copiant sur la carte SD) :

```
> dd if=/dev/block/mmc[mmc block number] of=/sdcard/mmc[mmc block number].img
```

Dans le cas où l'on souhaite réaliser un *dump* complet en se passant de l'utilisation d'une carte SD externe (tous les terminaux ne disposent pas d'un tel emplacement ou bien la carte SD n'est pas de taille suffisante), il est possible de réaliser une redirection TCP avec **adb**. Cette solution nécessite cependant d'avoir *Busybox* installé sur le terminal.

```
> adb forward tcp:5555 tcp:5555
> adb shell
> su /system/xbin/busybox nc -l -p 5555 -e /system/xbin/busybox
dd if=/dev/block/mmcblk0p30
```

Puis dans un autre terminal sur l'hôte de l'analyste :

```
> adb forward tcp:5555 tcp:5555
> cd /path/to/store/the/backup
> nc 127.0.0.1 5555 | dd of=mmc[mmc block number].img
```

D'autres méthodes sont décrites sur le *thread* XDA suivant [12].

4.3.5 ADBFS

Une dernière solution pour faciliter l'acquisition est d'utiliser un système de fichiers Fuse comme *adbfs* [8] pour monter via ADB le système de fichiers du terminal et en faire une copie.

Cependant, cette solution est évidemment moins optimale que la précédente, puisqu'elle ne permet pas de faire une copie bit à bit du système de fichiers. À noter que pour pouvoir parcourir le système de fichiers en tant que *root* (et accéder à des répertoires comme **/data/data**), il est nécessaire de mettre à jour les sources afin que chaque commande **adb** soit exécutée en tant que *root* grâce à la commande '**su -c**'.

4.3.6 Acquisition de la mémoire vive

Les terminaux Android ont deux types de mémoire : la mémoire volatile (ou RAM) et la mémoire non volatile (flash NAND) servant au stockage des données (identique à un disque dur).

Cette étape de l'acquisition de données consiste donc à *dumper* la RAM du terminal. Plusieurs solutions sont possibles :

1. Avant la version 2.3 d'Android, l'envoi d'un signal SIGUSR1 (kill -10 PID) permettait le *dump* de processus au format *hprof* (les *dumps* étant enregistrés dans le répertoire **/data/misc**).
2. Après la version 2.3 d'Android, pour *dumper* la mémoire d'une application, il est possible de passer par DDMS (*Dalvik Debug Monitor Server*), soit fourni dans le SDK avec l'outil monitor (l'outil **ddms** étant obsolète), soit avec *Eclipse* ou encore en utilisant l'*Activity Manager* avec **adb** :

```
> am dumpheap -n <pid> <dump file>
```

Seulement, pour pouvoir réaliser ce type d'action, il est nécessaire que l'application soit compilée en mode *debug*. Il est évident que dans le cadre d'une réponse à incident, cette condition limite fortement l'intérêt d'utiliser DDMS.



3. Une autre solution est d'utiliser l'API *ptrace* disponible sur Android avec les informations *procs* pour déterminer les zones de mémoire allouées aux processus afin de les récupérer.
4. La dernière solution reste de *dumper* la totalité de la mémoire en utilisant le module noyau Lime. Un tutoriel complet est disponible sur la page Google Project de Volatility [5].

Dans le cas de terminaux Samsung, la protection Knox [6] peut être présente et activée. Cette technologie permet de :

- vérifier l'intégrité du firmware ;
- appliquer des règles SELinux sur les processus Android ;
- utiliser un système de TPM avec les instructions TIMA du processeur ARM ;
- cloisonner les processus.

Les binaires **suid** tels que **su** ne possèdent pas la permission **CAP_SYS_SETUID**, empêchant alors leur exécution en tant que **root**. De même, SELinux se met en « *setenforce* » dès le *boot* du téléphone. Il est alors impossible d'ajouter des *capabilities* à de nouveaux binaires après le *boot*. Pour terminer, la protection Knox empêche le chargement de module noyau non vérifié. Elle vérifie régulièrement que le *kernel* n'a pas changé à partir des mesures obtenues auprès du *kernel* et en les comparant à celles du *kernel* d'usine d'origine. Elle authentifie les modules de *kernel* au fur et à mesure qu'ils sont chargés de façon dynamique.

Il est malgré tout possible de contourner Knox, car Odin ne restreint pas le flashage d'un nouveau noyau signé. Le **/system** peut donc être modifié afin d'ajouter des binaires ou de modifier des scripts de démarrage du mobile. Il est même possible de flasher un nouveau noyau n'appliquant pas les règles SELinux.

Mais à nouveau, utiliser Odin implique un *reboot* du terminal et réduit par conséquent l'intérêt d'acquérir la mémoire du terminal.

4.3.7 Déverrouillage des terminaux

Il est rare dans le cadre d'une réponse à incident que l'analyste ne puisse déverrouiller le terminal. Si malgré tout, il s'avère que ce ne soit pas le cas, différentes techniques existent selon que le terminal est rooté ou non.

Si le terminal est non *rooté*, l'attaque de Smudge [1] permet, si l'accès à la session du terminal est réalisé par un motif de déverrouillage, de reconstruire potentiellement le motif. Sur des versions anciennes d'Android, un chercheur a également mis en évidence la possibilité de débloquent l'accès au terminal en installant une application depuis le Google Play qui dispose de l'*intent* **android.intent.action.PACKAGE_ADDED**. Si le terminal est *rooté*, il est

possible de désactiver le verrouillage. Il existe en effet sous Android deux types de verrouillages, par schéma et par mot de passe. Les fichiers contenant le secret de ces verrouillages sont dans **/data/system** :

```
root@jflte:/data/system # pwd
/data/system
root@jflte:/data/system # ls -al *.key
-rw----- system system 0 2013-12-03 11:51 gesture.key
-rw----- system system 40 2013-12-03 11:51 password.key
```

En vidant le contenu du fichier **password.key**, la protection par code PIN du terminal est supprimée. Il en est de même avec le fichier **gesture.key** si le verrouillage se fait par motif.

Le déverrouillage peut aussi être bloquant lorsque la version d'Android est supérieure à la version 4.2.2. Une protection a été implémentée pour éviter l'exécution de commande **adb** lorsque l'hôte n'est pas un hôte de confiance. Il est théoriquement possible de réaliser cette action que si le terminal n'est plus verrouillé. Cependant, une vulnérabilité a été mise en évidence permettant de réaliser cette opération depuis le menu d'appel d'urgence [3].

4.3.8 Cas particulier des téléphones chiffrés

La plupart des ROM Android ne fournissent pas d'accès au terminal chiffré tel qu'un *escape shell* lors de la demande de *passphrase*. Cependant, comme dans le cas des terminaux non-chiffrés il nous faut obtenir un accès au terminal. De la même manière, il est possible d'obtenir cet accès si le terminal est allumé et non verrouillé ou avec le démon **adb** actif. L'autre solution est de se servir d'un *bootloader* non verrouillé pour obtenir l'accès aux données chiffrées. Le blog de Sogeti Esec couvre le cas particulier du HTC One [4].

5 Analyse des données

5.1 Analyse de la mémoire

L'analyse de la mémoire permet de détecter des techniques de compromission tout en mémoire comme l'injection de code dans des processus ou la création de processus cachés. Elle permet dans l'absolu de récupérer des informations dynamiques non présentes sur le système de fichiers.

Android étant basé sur Linux, l'analyse de la mémoire peut être effectuée tout simplement avec Volatility sous réserve d'avoir le bon profil. La page AndroidMemoryForensics [4] décrit la procédure pour créer ce profil. Il y a ensuite suffisamment de publications sur Internet au sujet de Volatility pour extraire les informations souhaitées.

5.2 Analyse du système de fichiers

L'analyse du système de fichiers permet dans un premier temps d'établir une chronologie des accès MAC aux fichiers, de récupérer les fichiers effacés, et de détecter d'éventuelles portes dérobées. Cette *timeline* doit permettre de reproduire toute l'activité de l'attaquant sur le système.

Sur les dernières versions d'Android, les systèmes de fichiers sont au format YAFFS2 ou Ext4. Sur les versions plus anciennes, nous pouvons retrouver les systèmes YAFFS, Ext3 ou encore Vfat.

L'analyse du système de fichiers s'effectuera donc classiquement avec sleuthkit [10].

5.3 Analyse des applications

5.3.1 Données des applications

Les bases de données

Les applications stockent leurs données sous la forme de fichiers XML, de bases de données au format *SQLite* ou plus rarement, de fichiers texte à plat. Ces informations sont accessibles essentiellement dans le répertoire **/data/data/<application>/databases/**.

La multiplicité des constructeurs et des versions d'Android est tout autant impactante à ce stade. Par défaut, les plus intéressantes dans le cadre d'une réponse à incident pourraient être les suivantes (la liste étant non exhaustive) : voir tableau ci-dessous.

Un simple client *sqlite3* suffit à analyser ces bases de données.

Pour illustrer un cas particulier, reprenons par exemple notre Samsung S4 avec Knox présent. Sur ce dernier, les données concernant l'historique du navigateur sont présentes essentiellement dans le fichier **/data/data/com.sec.android.app.sbrowser/app_sbrowser/Default/History**.

Les contentProviders

Les *contentProviders*, non évoqués jusqu'à maintenant, peuvent être tout aussi utiles que les bases de données

pour obtenir des informations intéressantes dans le cadre d'une réponse à incident.

Un *contentProvider* sert à stocker et récupérer des données et ainsi les rendre accessibles à toutes les applications. C'est le moyen le plus connu de partager des données entre différentes applications. Ils sont généralement décrits dans le *Manifest* du package :

```
<provider android:name="com.sec.android.app.sbrowser.provider.BookmarkWidgetProvider" android:exported="false" android:authorities="com.sec.android.app.sbrowser.widget" />
```

Un *contentProvider* se compose d'une URI et de méthodes (*Insert*, *Update*, *Delete*, *Query*). L'URI est de la forme :

```
<standard_prefix>://<authority content>/<data_path>/<id>
```

adb peut permettre de les interroger et de récupérer les informations. Par exemple, pour obtenir l'historique du navigateur, la commande est la suivante :

```
> dumpsys package providers // liste les contentProviders
> content query --uri content://com.sec.android.app.sbrowser.browser/history
```

5.3.2 Analyse du comportement des applications

Pour rappel, les applications sont de manière générale présentes dans les répertoires **/data/app** et **/system/app**. L'analyse d'applications Android est largement documentée sur Internet et les outils disponibles pour reverser les applications le sont tout autant :

- APKTOOL : <https://code.google.com/p/android-apktool/> ;
- Androguard : <http://androguard.blogspot.fr/> ;
- JEB : <http://www.android-decompiler.com/> ;
- DroidBox : <https://code.google.com/p/droidbox/>.

L'analyse d'une application se décompose en deux phases :

1. Analyse statique :
 - analyse du code binaire et traduction en code java, consultable depuis l'applicatif (utile aussi pour l'analyse de l'obfuscation du code) ;
 - récupération des éventuels conteneurs de certificats pour fabrication de l'interception SSL (pour analyse dynamique ultérieure) ;

Cookies, Formulaires remplis	/data/data/com.android.browser/databases/webview.db
Cache des adresses parcourues	/data/data/com.android.browser/databases/webviewCache.db
Fichiers téléchargés	/data/data/com.android.providers.downloads/databases/downloads.db
Applications installées du market	/data/data/com.android.vending/databases/assets.db
Favoris/recherche browser	/data/data/com.android.browser/databases/browser.db
Fichiers téléchargés	/data/data/com.android.providers.downloads/databases/downloads.db



- analyse des fichiers temporaires stockés sur Android et notification lors de leur création, suppression, modification (pour analyse statique ultérieure) ;
- audit de code focalisé sur les classes dangereuses et vérification des permissions.

2. Analyse dynamique :

- récupération des journaux du téléphone ;
- récupération du contenu de la *heap* (la mémoire allouée dynamiquement lors de l'exécution du programme) pour faire de l'introspection sur son contenu ;
- écoute du trafic réseau en direction et depuis Internet ;
- usurpation du certificat client, écoute et réinjection par un proxy avec le certificat de l'application.

Il est important de ne pas négliger l'analyse réseau de l'application. Le comportement par défaut des solutions actuelles se concentrant trop sur l'analyse du code binaire et du comportement à l'intérieur du terminal.

6 Industrialisation

À plusieurs reprises a été évoquée la complexité de l'analyse forensics à cause des différents constructeurs et versions d'Android.

Dans le cadre d'une réponse à incident dans une entreprise, cette analyse peut être rendue plus simple et peut alors être industrialisée selon que cette entreprise est mature ou non au niveau de sa gestion de flotte mobile.

Si ce n'est pas le cas, elle sera alors contrainte à autant de procédures d'acquisition de données qu'il y a de terminaux et de versions d'Android différents utilisés par ses collaborateurs dans un cadre professionnel.

Si la société a pris au sérieux l'extension de son SI sur des terminaux mobiles en « imposant » un modèle de terminal et d'OS, des solutions automatisées d'investigation s'offrent alors à elle.

6.1 Homogénéité des terminaux

L'homogénéité des terminaux que ce soit au niveau du constructeur, de la version du système d'exploitation ou encore des applications installées, facilitera bien évidemment la définition de procédures et de méthodologies génériques d'investigation au sein de l'entreprise.

Cette homogénéité permettra à l'entreprise d'une part de définir un référentiel sur lequel se baseront les analyses forensics et d'autre part de développer et de maintenir ses propres outils d'analyse forensics.

6.2 Un mot sur les solutions MDM

Les solutions MDM (*Mobile Device Management*) sont de plus en plus présentes au sein des entreprises. MobileIron et Airwatch sont les plus connues. Elles ont pour objectif de pouvoir gérer la flotte professionnelle et d'assurer les fonctionnalités suivantes :

- accéder rapidement aux e-mails, agendas et contacts professionnels ;
- accéder automatiquement aux réseaux Wi-Fi et VPN de la société ;
- rechercher et installer facilement les applications professionnelles ;
- vérifier l'application des règles de sécurité de la société ;
- localiser les appareils perdus ou volés.

Même si ces solutions sont (censées ?) apporter une couche de sécurité supplémentaire, elles n'apportent malheureusement à notre connaissance aucune fonctionnalité utile à une réponse à incident. Elles ne permettent pas typiquement de créer une copie bit à bit de la mémoire ou du stockage, ou encore d'exécuter des commandes à distance sur les terminaux.

6.3 Automatisation des acquisitions et des analyses

L'automatisation de l'acquisition et de l'analyse des mobiles est multiple :

- elle permet de s'assurer que les méthodologies sont identiques quel que soit le téléphone concerné ;
- elle est nécessaire quand le nombre de mobiles à analyser est conséquent (dans un contexte d'entreprise) ;
- elle facilite la mise à jour des méthodologies et processus d'analyse.

6.3.1 Acquisition des données

La majorité des données système acquises le sont à travers **adb**. Il est donc assez simple de développer à minima des scripts pour acquérir automatiquement d'une part les logs et d'autre part l'environnement à investiguer (la liste des packages, leur configuration et celle de l'OS, etc.) et de comparer avec un référentiel défini au départ basé sur un « master ».

6.3.1 Analyse des applications

L'analyse des applications au sein du téléphone peut s'avérer fastidieuse de par le nombre conséquent



d'applications à analyser et des techniques de compromission possibles (l'application Android est en effet le principal vecteur d'attaque). L'industrialisation de l'analyse des applications est donc primordiale. Elle doit permettre l'analyse statique et dynamique d'une application définie précédemment.

Afin d'atteindre ces objectifs, nous pouvons imaginer une solution composée de plusieurs briques logicielles qu'il est ensuite possible de « scripter » pour automatiser les différentes phases d'analyse.

La première brique logicielle concerne l'émulateur Android de Google fourni avec le SDK comportant une ROM avec accès root et les outils d'analyses forensics nécessaires. Il a pour objectif de simuler le téléphone utilisateur. Il permet entre autres de répondre à différentes contraintes techniques telles que le routage des flux réseaux et l'installation ou contrôle automatique d'applications Android. Un agent se charge ensuite de la surveillance de l'application à analyser sur le téléphone. Il peut être développé en C avec le NDK, le SDK natif de Google. Cet agent se connecte au système par l'intermédiaire d'**adb**. Plusieurs techniques sont ensuite possibles pour surveiller l'application :

- l'API ptrace avec les informations du process pour déterminer les zones de mémoire allouées par l'application afin de les récupérer ;
- l'API inotify pour surveiller le système de fichiers et connaître l'évolution des accès MAC aux fichiers ;
- un système de hooks directement basé sur **LD_PRELOAD** (plutôt que la librairie bionic qui a parfois un comportement anormal) pour surveiller davantage que la gestion de la mémoire et les accès aux fichiers. Si la ROM est contrôlée, une dernière solution est d'activer le mode debug et d'utiliser le protocole JDWP pour *debugger* directement la partie Java des applications. Un proxy, situé en dehors de l'émulateur Android, s'ajoute au reste de notre solution. Il a pour objectif d'écouter tout le trafic de l'application Android vers et depuis Internet. Il est également contrôlable depuis le référentiel pour la mise en coupure des requêtes de l'appli ou simplement pour le « rejete » de requêtes.

La dernière brique logicielle concerne le référentiel. Il est la pièce maîtresse de la solution et permet de contrôler les différents éléments de l'application, l'émulateur, l'agent et le proxy. C'est au travers de ce référentiel que sont effectuées toutes les analyses forensics nécessaires :

- il propose une vue de l'application Android avec le code source régénéré avec Androguard et permet de faire des recherches dans le code généré ainsi que ses ressources texte.
- il permet également de modifier des éléments au sein de l'application, de la « re-signer » et de la réinstaller comme par exemple changer une clef RSA pour fausser les vérifications de signature.

- il permet de récupérer les éléments de l'agent tels que les noms de fichiers modifiés avec leur contenu et le log de l'émulateur, la mémoire du process, etc. Couplé avec la recherche intégrée du code régénéré d'Androguard, ce dispositif est utile pour identifier les éléments qui génèrent les traces observées. Ce référentiel a donc le contrôle total de l'émulateur via **adb** et l'agent. Il contrôle également le proxy afin qu'il puisse agir comme proxy passif ou actif à la manière de spike ou burp proxy. Il permet également la consultation des requêtes relayées par le proxy.

La mise en place d'une telle solution peut paraître complexe, mais la fréquence et la quantité des applications à analyser au sein d'une entreprise l'impose. Bien entendu, la solution proposée en est une parmi tant d'autres. Il a d'ailleurs été présenté au SSTIC 2014 la solution Hooker [13], une solution d'analyse automatisée de *markets* Android, preuve que nous sommes sur la bonne voie :)

Conclusion

Bien qu'un seul article ne suffise pas à traiter la problématique de réponse à incident sur Android, nous avons néanmoins abordé « succinctement » les différentes techniques d'acquisition selon les constructeurs et l'environnement à acquérir. Le sujet de l'industrialisation mériterait l'écriture d'un article à lui tout seul et il est évident, de par la diversité des terminaux et du faible outillage existant, que le sujet de la réponse à incident sur Android, et sur mobile de manière générale, est un sujet complexe qui n'en est qu'à ses débuts. ■

■ RÉFÉRENCES

- [1] ABE : <https://github.com/nelenkov/android-backup-extractor>
- [2] ODIN : <http://odindownload.com/>
- [3] TOWELROOT : <https://towelroot.com/>
- [4] LIME : http://code.google.com/p/volatility/wiki/AndroidMemoryForensics#Load_the_LIME_LKM_on_the_Android_Device
- [5] KNOX : <http://www.samsung.com/fr/business/solutions-services/mobile-solutions/security/samsung-knox>
- [6] ADBFS : <https://github.com/spion/adbfs-rootless>
- [7] SMUDGE : http://en.wikipedia.org/wiki/Smudge_attack
- [8] URGENCE : <https://labs.mwrinfosecurity.com/advisories/2014/07/03/android-4-4-2-secure-usb-debugging-bypass/>
- [9] HTC One bootloader : <http://esec-lab.sogeti.com/post/Exploiting-a-vulnerability-in-HTC-One-bootloader-and-bruteforcing-the-PIN-password>
- [10] VOLATILITY : <http://code.google.com/p/volatility/>
- [11] SLEUTHKIT : <http://www.sleuthkit.org/>
- [12] XDA : <http://forum.xda-developers.com/showthread.php?t=1818321>



LA CITADELLE EST TOMBÉE ! CYBERCRISE, COMMENT S'ORGANISER ?

Sébastien Bombal - sebastien.bombal@orange.com

Orange Cyberdefense - Directeur des Opérations

mots-clés : EXFILTRATION DE DONNÉES / APT / COMPROMISSION / GESTION DE CRISE / ORGANISATION

«

Exfiltration de données », « APT », « compromission », pourquoi tant d'affaires ? Quelles en sont les causes ? Comment gérer une telle situation ?

1

Vous connaîtrez la crise... mais laquelle ?

En juillet 2012, le rapport du sénateur Bockel [1] soulignait la prise de conscience des politiques d'un « fléau », déjà sous les projecteurs des médias : les compromissions informatiques de grandes sociétés ou d'administrations françaises.

Phénomène international, depuis 2010 les affaires éclatent, se succèdent et se ressemblent. PME au patrimoine sensible, opérateurs d'importance vitale, grands groupes, administrations, nul n'est épargné.

Il est nécessaire d'avoir un **regard humble** sur toutes ces affaires, car elles peuvent soudainement et même brutalement vous embarquer. Ne l'oublions pas, la gestion des incidents et des crises fait partie de notre métier de la sécurité.

Alors comment toutes ces affaires de vol de données deviennent-elles possibles ? Pourquoi n'est-il pas aisé de remédier, voire d'éviter toutes ces crises ? Que font ces organisations qui découvrent une attaque réussie, des vols ou diffusions d'informations sur Internet ? Pourquoi découvrent-elles tardivement que leur système d'information est contrôlé par une tierce personne ou entité ? Comment faire face lorsque vous êtes responsable de la sécurité d'un SI compromis ?

Autant de questions que nous allons essayer de traiter dans cet article. Si la littérature des bonnes pratiques pour protéger ou renforcer est abondante, il n'en est rien de la littérature pour détecter ou remédier à une attaque.

Cet article est donc un retour d'expérience empirique de crises traitées pour différents secteurs d'activités.

2

Les grandes phases d'une attaque versus la détection

2.1 L'intrusion

Compromettre un système d'information sera toujours plus facile que de le défendre. Attaquer une cible revient toujours à rechercher un arbre d'attaque, exploitant son ou ses maillons faibles. Que ce maillon soit organisationnel ou technique, il suffit de trouver l'endroit où les boucliers ne sont pas présents, ou juste plus vulnérables.

Une attaque se compose souvent d'une action d'intrusion sur un système informatique à distance ou grâce à un accès interne au système d'information. Les voies les plus exploitées sont les codes malveillants via des médias amovibles de type clés USB, les mails malicieux, des sites Internet déjà compromis, l'exploitation de vulnérabilités de logiciels ou encore l'ingénierie sociale.

Malheureusement [2], les modèles d'organisations (matricielles, par projet, fonctionnelles...) contribuent à créer des brèches. Gérer la sécurité d'une société est loin d'être aisé contrairement à ce que peut attendre un individu. Prenons quelques exemples de cette complexité :



- Des organisations multiculturelles, internationales, dispersées aux quatre coins de la planète entraînant des difficultés organisationnelles de coordination et de contrôle ;
- Un appel à la sous-traitance de plus en plus massif, qui elle-même utilise des sous-traitants (*cascading*) ;
- Des fournisseurs et partenaires de plus en plus intégrés aux systèmes d'informations (*supply chain*, globalisation...) ;
- Des clients de plus en plus connectés au SI également pour des services à valeurs ajoutées ;
- Des réglementations inconsistantes entre pays, et des autorités de plus en plus connectées aux entreprises ;
- Des filiales, joint-ventures, plateaux projets dont la structure complexifie les rôles et les responsabilités et obligations de chacun ;
- Des transformations IT ou organisationnelles qui s'étalent sur des années, de plus en plus nombreuses et rarement achevées à 100 % ;
- Une digitalisation de certains métiers qui n'ont pas pris en compte la cybersécurité (systèmes industriels, gestion technique de bâtiments GTB/BMS, objets connectés...) ;
- Des changements de périmètre fréquents (déploiement à l'étranger, acquisition/cession de société, démantèlement de sites pour se repositionner, changement d'organisation interne...).

Sans compter qu'il n'est pas rare d'avoir dans les parties prenantes un acteur qui serait à la fois concurrent de vos activités sur certains marchés, fournisseur ou sous-traitant pour d'autres, mais aussi partenaire pour du business (alliance, projets communs), ou qu'il soit un client et même parfois votre investisseur/actionnaire...

Autant de vecteurs possibles organisationnels exploitables pour pénétrer un SI. Qui de ces maillons est le plus faible ? S'introduire via un partenaire est un vecteur potentiel, alors pourquoi ne pas l'exploiter ? Pourquoi ne pas passer via votre infogérant connecté aussi à de multiples clients ? Ou même via une autorité, autorité connectée aux entreprises d'un même secteur d'activité pour des besoins de contrôle ?

Avec la complexité et la dépendance digitale aux technologies, les grandes organisations ne sont pas structurées pour empêcher ou faire face à une cybercrise. Et si on couple cela au fait que la technologie est de plus en plus complexe, de plus en plus intégrée, virtualisée, évolutive et rapidement obsolète, le nombre de vulnérabilités devient exponentiel.

Prenons une technologie au hasard : VSPHERE. Le guide de sécurisation éditeur de VSPHERE [3] fait ~150 mesures de sécurité à déployer après installation. Pour un grand parc de serveurs (1000, 5000, 10000

ou même plus), il est obligatoire d'industrialiser le déploiement. Il est impossible d'employer la méthode dite du « *runner* », administrateur qui va passer sur les serveurs, pour appliquer les 150 mesures. Le risque d'erreur ou encore le facteur temps rend la méthode inadaptée.

Alors l'industrialisation va devoir être menée sur un périmètre (connu), soit de fait excluant la partie inconnue du parc. L'organisation va mesurer avec un indicateur de déploiement jusqu'à un niveau « acceptable ». 80 %, 85 %, 90 %... Et 90 % de 10000, ça fait toujours 1000 serveurs vulnérables, plus les serveurs non connus dans l'organisation du déploiement...

Ajouter quelques correctifs de sécurité manquants, quelques incompatibilités d'applications, quelques changements de périmètres en cours de route, quelques systèmes obsolètes et les maillons faibles se créent, se multiplient... Sans parler d'un risque de 0 Day. Pourtant des efforts et du temps sont consacrés par ces structures pour atteindre 80-85-90%. Mais est-ce suffisant pour de la sécurité face à de l'espionnage de plus en plus massif ?

Il y a comme un décalage entre des capacités d'espionnage de plus en plus simples et une sécurité basée et objectivée sur des moyens/indicateurs, et pas sur un résultat, un effet recherché.

En conclusion, même si on se doit de dissuader, protéger et même prévenir des risques d'intrusion, à grand renfort de systèmes de management, d'audits, de bonnes pratiques, de contrats et de politiques de sécurité... cela reste insuffisant pour empêcher les intrusions.

Une intrusion se fera. Et comme on ne peut pas tout surveiller non plus pour des questions de réglementation, de volumes de données, de coûts, d'exhaustivité de périmètre, d'organisation... l'effort de détection ne doit pas être sur le moment de l'intrusion (ex. : le clic malheureux sur un code malveillant), mais sur les étapes d'après.

2.2 La propagation latérale et verticale

Une fois un point d'accroche compromis dans le système d'information, l'attaquant essaiera d'obtenir davantage de contrôle du système en essayant :

1. D'obtenir des **privilèges additionnels** ou la maîtrise de composants clés ;
2. De compromettre un maximum de **points d'accès** pour des entrées/sorties et un maximum de composants afin de renforcer sa **capacité de nuisance**.

L'attaquant n'exploitera probablement pas d'outils ou moyens exogènes sur votre SI. Il utilisera vos propres



outils et moyens pour compromettre plus de machines (propagation latérale) et obtenir plus de droits, jusqu'aux moyens d'administration (propagation verticale).

Ainsi les outils natifs (Windows, Unix, PMAD...) ou encore d'administration et d'exploitation deviennent les moyens de compromission.

La **propagation latérale** n'est pas trop grave en soi tant que le volume de systèmes/applications/comptes perdus reste à taille humaine ou traitable par votre organisation, lorsqu'il va falloir y remédier. Cela constituera l'ensemble des points d'entrées/sorties potentiels de votre attaquant qu'il faudra nettoyer de manière exhaustive.

En revanche, la **propagation verticale** est bien plus grave. C'est comme un cancer, si un point névralgique est atteint, c'est un point de non-retour, il faudra une opération lourde de reconstruction appelée « bascule » ou « pivot ». Une image médicale serait de construire un nouveau SI avec « un nouveau cœur et le transplanter dans le SI en ablatant les parties malades ».

Avant de parler du traitement de l'incident, quels sont ces composants critiques recherchés par les attaquants ? Tout simplement l'infrastructure IT (PKI, AD, annuaires/IAM, virtualisation, moyens de télédistribution, outils de sécurité, ordonnanceurs, comptes privilégiés d'administration, sauvegarde/restauration...).

Tous ces outils d'administration et composants sont la base du fonctionnement d'un SI. Ils ont une empreinte massive, privilégiée et consolidée.

Prendre le pouvoir sur ces composants, c'est être certain que l'organisation est complètement compromise, et avoir la capacité de nuisance maximale et de résilience dans la cible. On parle de **risque systémique**. Ce risque c'est l'effet domino. Prendre le contrôle d'un compte à haut privilège suffit pour que toute la sécurité tombe.

Cette notion de **risque systémique** est extrêmement importante pour la suite et le traitement de l'incident. L'explosion du nombre de ces incidents est aussi le résultat d'une incompréhension de la sécurité. Le marché est uniquement tourné vers les risques « métiers ». « Métiers » dites-moi ce qui est vraiment à sécuriser et ménageons nos efforts sur le reste... Mais qui va porter la responsabilité de ce risque systémique ? De cette chaîne de confiance avec un maillon faible ou de la sécurité des clés de voûte comme l'infrastructure, les moyens communs ou transversaux du SI ?

C'est souvent ce risque systémique non maîtrisé, porteur de vulnérabilités importantes, qui aggrave dramatiquement les compromissions.

Si on revient à la détection, il est souvent fait allusion aux **signaux forts** et **signaux faibles**.

1. Les signaux forts sont provoqués par les scénarios d'attaque que vous connaissez et maîtrisez.

Ce sont des attaques « connues », dont les moyens de détection et de blocage viennent souvent des fonctions techniques de sécurité ou des procédures de sécurité. Ces attaques génèrent un « bip » fort et sont souvent bloquées automatiquement.

Mais avec la problématique du volume, de l'exhaustivité du contrôle du SI et les écarts d'organisation, il n'est pas rare que l'intrusion passe et se propage sans mise en alerte.

2. Les attaques « inconnues » peuvent toutefois être détectées tôt, à défaut d'être bloquées. Avec des procédures, des moyens humains de type SOC, des analyses de traces, une connaissance du comportement du SI ou encore des outils plus « next-gen » de sécurité. On peut isoler des nouveaux signaux forts et lever des incidents.
3. En revanche le vrai signal faible, l'indétectable, ne sera trouvé que par deux moyens :
 - a. a posteriori une fois l'agression réussie et **l'impact vécu par une organisation**,
 - b. **le hasard**.

Ce dernier n'est pas à sous-estimer. Capter « le hasard » dans une organisation est loin d'être évident. Captez-vous, voire prêtez-vous attention à toutes ces petites anomalies sur le SI ? À cet administrateur s'étonnant d'un comportement anormal d'un compte ou d'un serveur ?

Les grandes comme les petites organisations disposent rarement de moyens opérationnels permettant de traiter les 3 étages de la détection ci-dessus pour détecter de manière efficiente et efficace une expansion.

Le **hasard** ou la **découverte de l'impact** (comme la publication de l'information sur Internet à l'image de Wikileaks) implique une découverte a posteriori, longtemps après l'intrusion. L'intrusion peut alors être profonde (verticale) et large (latérale), a même pu être nettoyée et mise en stand-by (dormante). Combien de temps met un attaquant pour me faire perdre le contrôle du SI, versus le temps qu'il me faut pour le détecter et le contenir ? On découvre ainsi un facteur primordial pour **mesurer l'efficacité de la sécurité : le facteur temps**.

Retrouver tous les points compromis pourra s'avérer long, fastidieux, probablement non exhaustif, voire même impossible. Combien de temps me faudra-t-il pour remédier à la situation et revenir à l'état nominal ? Peut-être des mois en fonction de l'expansion, des mois pendant lesquels vous allez devoir vivre avec un agresseur dans votre environnement.

2.3 L'objectif de l'agression

L'attaquant une fois les cibles de compromission atteintes peut procéder à la réalisation de son objectif qui peut être résumé à trois objectifs génériques :



- **Collecter et exfiltrer** de l'information à des fins de renseignements. Cela nécessite une connaissance du métier, de la langue et de la cible. Le renseignement peut être de 3 types :

- Le renseignement pour le cyberspace (renseignement technologique pour des actions potentielles ou à fin d'action liée à un dossier d'objectif) ;
- Le renseignement dans le cyberspace. CYBINT, renseignement de masse à l'image de l'affaire SNOWDEN ;
- Le renseignement par le cyberspace. Renseignement ciblé.

- Le **sabotage** ou **l'entrave** : détruire tout ou partie du système compromis ;
- **Altérer** ou **interagir** avec tout ou partie du système compromis. Peut servir des objectifs multiples tels que l'entrave, un rebond pour un autre objectif, faire de la guerre de l'information...
- La **subversion** ou la **déstabilisation temporaire** ou **permanente** d'une organisation.

Concernant les cas de vol de données, certaines entreprises mettent en place des mesures de détection a posteriori, via une veille spécialisée. L'idée est de détecter via l'impact une exfiltration non détectée, mais réussie.

Typiquement ce type de surveillance recherche des transactions/publications de données personnelles, de numéros de cartes bancaires, dossiers de R&D, etc., sur les *black markets*. Sans oublier la veille « Presse »...

3 Traitement type d'une cyberattaque, gestion d'une cybercrise

3.1 Les grandes étapes

Partons du principe qu'un capteur technique ou organisationnel initie le déclenchement d'une cellule de crise.

Il a **3 périodes majeures** dans une crise de ce type.

La première constitue la période de **compréhension de l'attaque**. Cette période doit aussi permettre de commencer à renforcer le système d'information pour éviter de nouvelles voies de compromission et de préparer une étape clé pour la reprise en main du contrôle du système d'information.

Pour comprendre et renforcer, quatre types d'actions doivent être menées de concert :

- le *post mortem* ;
- le *live forensics* ;

- la cartographie de votre SI ;
- l'audit.

Les analyses **forensics** s'alimentent de manière croisée, quels outils, quels moyens, quels canaux de contrôle, commandes d'infiltration, d'exfiltration, d'entrave utilisent l'attaquant. Le live le suit à la trace en « temps réel » là où le post mortem recherche dans le passé.

La **cartographie** et **l'audit** sont là pour donner une vision exhaustive du scope et des vulnérabilités. C'est un travail conséquent à mener pour éviter un « sur accident », ou une réinfection à la fin de l'opération. Si la cartographie, l'audit et les analyses forensics **ne sont pas exhaustifs et parfaits, l'opération risque l'échec**.

Toutes les mesures identifiées de remédiation ou de corrections de vulnérabilités devront être classées dans un **plan et priorisées** :

1. Les **mesures déployables tout de suite** et qui n'éveilleront pas de soupçons de votre activité (montée de version logicielle, patch de sécurité...) et la préparation des systèmes sains pour la phase 2 ;
2. Les **mesures à déployer sur la phase 2** (mesures de prérequis comme l'isolation, mesures de nettoyage/remédiation, mesures de renforcement nécessaires et qui auraient éveillé les soupçons de l'attaquant en phase 1) ;
3. Les **mesures à déployer post gestion de crise** pour continuer le renforcement et maintenir l'attaquant en dehors du SI.

La **deuxième période** est la « **bascule** ». C'est l'étape de reprise en main du système, soit l'étape où les systèmes compromis sont remplacés par des systèmes sains. Cette étape implique souvent, une isolation du périmètre avec l'extérieur de tout type, un arrêt partiel ou complet du système compromis et des systèmes dépendants, puis une reconstruction partielle ou totale avant la réactivation de l'ensemble des services. Des mesures de renforcement additionnelles sont déployées aussi pendant cette phase.

Cette étape est très courte sur l'échelle de temps. Quelques heures ou jours. Son succès dépend de la qualité de la préparation de la première période, de l'exhaustivité de la liste des systèmes compromis à remédier et d'une coordination de crise parfaite.

Il faut être conscient qu'arrêter un système d'information (« switch off ») est facile. Le redémarrer est une autre affaire et les impacts vont être nombreux.

La **phase 3 « post bascule »**, est l'étape où :

1. on gère les impacts de la phase 2 ;
2. on continue de déployer les mesures de sécurité prévues ;
3. on surveille l'attaquant et le SI pour vérifier que l'opération est un succès.

En résumé, l'étape 1 contient :

- Audit des vulnérabilités : cette phase est l'étape permettant de recenser toutes les vulnérabilités du système d'information ciblé et de prioriser ;
- Audit de la nature de la compromission ;
- Préparation d'une bascule, reprise en main ;
- Déploiement des mesures de sécurité pré-bascule.

L'étape 2 se concentre sur la reprise en main de la situation avant de laisser place à une dernière étape, l'étape 3 qui consiste à maintenir l'effort de renforcement.

Un découpage alternatif des tâches peut être synthétisé sous la forme suivante :

- Veille et surveillance ;
- Détection et traque ;
- Identification ;
- Réaction ;
- Restauration ;
- Gestion des impacts, renforcement et évaluation du succès.

C'est un programme de transformation « sécurité » pour l'entreprise, mais qui doit être mené avec une direction de crise.

Le **tempo** et la prise de décision sont primordiaux, la non-décision est catastrophique. La chaîne de commandement doit être claire et efficace. Toutefois, transformer une **posture de défense** rapidement avec un objectif de résultat (aucune intrusion) se résume toujours à 4 mots :

- **Systémique** : sécuriser et reprendre le contrôle des composants critiques pour supprimer le risque effet domino ;
- **Systématique** : identifier et corriger les systèmes compromis (canaux d'entrées/sorties et moyens de nuisance) de manière exhaustive, mais aussi corriger les vulnérabilités permettant un retour de l'attaquant de manière tout aussi exhaustive ;
- **Efficacité** : Il n'y a plus de logique de moyens, mais un objectif de résultats. C'est la transformation à opérer pour les organisations ;
- **Efficience** : il faut être pragmatique pour l'efficacité des mesures, il n'est pas question d'avoir des mesures de sécurité inapplicables, ni d'avoir une armée mexicaine pour pouvoir l'appliquer. L'usage des ressources doit rester juste et nécessaire, pour s'inscrire dans la durée et conserver l'efficacité.

3.2 Le sponsorship et le seuil de crise

La direction de crise, la remédiation de la situation et la transformation de la posture de sécurité pour ne plus jamais à subir cette situation, nécessite le **plus**

haut niveau de sponsorship de votre organisation. PDG et DG sont le minimum.

Cela ne veut pas dire qu'ils doivent participer de manière opérationnelle, mais ils seront **parties prenantes** pour :

- Monopoliser les ressources adéquates et donner LA priorité sur ce dossier.
- Permettre le fonctionnement en dehors des processus habituels comme toute crise. Mais attention, on vous demandera des comptes après la crise des emplois/ressources et choix faits.
- Adouber les arbitrages d'impacts sur le business. Fermeture de la connexion internet pendant X jours ? Bascule de la production en manuel ou même arrêt de l'usine X ? Fermeture définitive jusqu'à nouvel ordre du service X aux clients ? Aux collaborateurs ?
- Porter les messages sécurité et faciliter la conduite du changement aux managers dans l'entreprise.
- Être au courant des impacts de l'attaquant (exfiltrations en cours, à venir, déjà effectuées, sabotage en préparation...) et entériner le déclenchement des opérations de remédiation.

L'argent dans ce genre de cas est rarement un problème a priori, mais potentiellement a posteriori. Vous devez donc tracer toutes les activités, décisions, observations et éléments de décisions lors de la crise. Management, audit interne, actionnaires, clients, partenaires, presse et même autorités pourront être amenés à vous poser beaucoup de questions. Après la fin du traitement des impacts, il y a cet « **après crise** ». Et il n'est pas à sous-estimer.

Alors **quand déclencher** ce seuil de crise ? Il y a au moins deux cas simples :

- **L'impact expose vos dirigeants** (juridiquement, médiatiquement, financièrement...) ;
- Vous n'êtes plus le seul maître à bord, le **risque systémique** vient de se concrétiser.

3.3 Quelles sont les tactiques possibles lors d'une cybercrise vis-à-vis de l'attaquant ?

Il faut d'abord évaluer des critères primordiaux :

- l'attaquant sait-il que vous l'avez découvert ?
- l'affaire est elle toujours confidentielle ?

Commençons par le cas simple : « non, l'attaquant ne le sait pas ». Vous avez donc l'**initiative** au moins jusqu'à la « bascule ». L'isolation complète du SI et la suppression des moyens qu'il utilisait à travers le déroulement du plan de la bascule va informer l'attaquant



qu'il a été repéré et que vous tentez d'y remédier. Que la bascule réussisse ou échoue, vous n'avez plus l'avantage de l'initiative à la fin de l'opération.

La bascule peut échouer si les canaux d'entrées/sorties et autres moyens de nuisance de l'attaquant ne sont pas identifiés et traités de manière exhaustive.

Il peut être intéressant d'utiliser votre avantage de l'initiative pour travailler sur la menace. Les tactiques défensives autour de l'information existent depuis longtemps (désinformation, intoxication informationnelle, etc.). Prenons un exemple, si un intrus s'intéresse à vos numéros de cartes bancaires, pourquoi ne pas lui laisser essayer de voler et de revendre des faux numéros, le temps de traiter l'intrusion ?

Des tactiques sont possibles, c'est un volet du renseignement que l'on oublie. Si l'espionnage est à la portée de tous maintenant via le cyber, certaines techniques de contre-espionnage via le cyber ne le sont pas moins.

Deuxième cas, l'attaquant sait qu'il est démasqué. À partir de ce moment-là, il faut prendre une décision sur une **tactique de harcèlement**. Il est souhaitable de mettre des moyens à l'encontre de son agresseur qui vont rendre sa tâche quotidienne d'infiltration/exploitation difficile. Il se connecte avec un compte, suppression du compte, il se connecte depuis une IP externe, blacklistage, etc. Toutes ces petites mesures qui complexifient et l'obligent à griller ses cartouches rapidement. En revanche, vous entrez dans un risque de représailles dans votre SI, à l'image d'une guérilla. Il peut lui aussi vous entraver pour vous retarder.

L'autre avantage tactique dont il faut **retarder la perte** (car vous la perdrez tôt ou tard), c'est la **confidentialité** de l'affaire. Travailler en confidentialité vous évite de consommer des moyens et de subir une pression supplémentaire (presse, clients, partenaires...). Ces parties-prenantes vous demanderont des comptes, informations et engagements tôt ou tard.

En revanche, il faut être prêt à tout moment et avoir une ligne de communication externe et interne adaptée. Ne négligez pas l'interne, c'est une source critique de commentaires vis-à-vis de l'extérieur.

Perdre la confidentialité implique la perte quasi systématique de l'initiative.

Et tenez-vous prêt à dérouler le « plan » de remédiation à tout moment, même si la phase de pré-bascule est incomplète. Il est possible que l'attaquant atteigne un point insoutenable pour votre organisation.

3.4 Les autorités

J'alerte ? J'attends ? Je passe sous silence ? Vu du citoyen, du consommateur ou d'un état c'est évidemment la transparence qui est demandée aux organisations, à travers l'obligation de notification des incidents.

C'est souvent perçu comme si vous étiez mal garé et que vous deviez héler la pervenche pour demander une amende.

En France par exemple, dans le secteur télécom pour les données personnelles, il est obligatoire de notifier la CNIL dès la détection de l'impact sur les données personnelles. Mais ce n'est pas le cas de tous les secteurs. Dans le cadre de la LPM (Loi de Programmation Militaire) les obligations de notification pour les OIV (opérateurs d'importance vitale) vont être élargies.

Quoi déclarer ? Quand le déclarer ? C'est toujours une question difficile des deux côtés : autorités pour légiférer et les organisations qui subissent. Mais vu des entreprises, il faut le faire en se posant deux questions :

- Quelles sont mes obligations et que peut m'apporter l'autorité ? En Europe, on a généralement plus à y gagner à les notifier très vite et à bénéficier d'appuis.
- Quand vais-je perdre la confidentialité de cette affaire ? Si l'affaire va devenir publique, ou que vous avez des parties prenantes (fournisseurs, sous-traitants...) concernées par la crise, ne vous posez pas de question. Vous ne tiendrez pas longtemps la confidentialité : notifiez-les.

3.5 Quelles sont les constantes d'une cybercrise ?

Une des constantes lors d'une cybercrise, c'est le besoin de faire appel en un temps très court, pour une période potentiellement longue, à un volume important de ressources humaines pour participer aux différentes étapes de résolution d'une crise.

Toutes les étapes décrites précédemment prennent du temps et nécessitent un nombre important de ressources humaines avec une gestion adéquate (roulement des ressources, gestion de la coordination, planification des tâches, etc.). Lors d'une attaque, les notions de 35 h, jours ouvrés, vacances et RTT n'ont plus vraiment de sens.

Le Code du travail, n'est clairement pas adapté à la notion de cyberattaque. La gestion de ressources lors d'une crise est déjà une difficulté, mais pour ce type de crise c'est encore plus vrai à cause du facteur temps et le faible nombre d'experts. Faites vous aider sur cet aspect et anticipez ce point.

Ces étapes nécessitent des compétences variées, classables en 3 profils :

- **coordinateur** : management de crise ;

- **l'expert** cybersécurité : identifier et travailler sur les mesures techniques à mettre en œuvre ;
- **profils d'exécutions** informatiques : déployer les mesures techniques et procédures préparées par les experts.

Où trouver ce réservoir potentiel de ressources ? Le marché peut le fournir dans une faible proportion.

À la date d'écriture de cet article, l'ANSSI vient de publier un « Appel public à commentaires sur le référentiel d'exigences applicables aux prestataires de réponse aux incidents de sécurité (**PRIS**) » [4], à l'image du référentiel d'exigences PASSI pour les prestataires d'audits.

Mais les systèmes d'information des entreprises utilisent des technologies complexes (ERP, PLM, GED, collaboratif, mainframe...) et parfois des versions obsolètes de celles-ci. Où trouver des compétences z/OS d'un mainframe dans l'urgence ?

Cette question des compétences et du nombre est primordiale lors de la préparation des crises de ce type. À l'échelle d'un pays, on cumule en plus la problématique de couverture géographique et d'urgence. Des experts sur une technologie X ? Oui à Paris, Lyon, Lille, Nantes... et les autres villes ?

Toutes ces ressources, pour une longue période, avec des spécialités en sécurité ou sur une technologie font tourner chaque jour très vite le compteur financier ! Le coût d'une affaire avoisine vite les 10 M€ sur un grand parc, montant que l'on peut couvrir maintenant avec des polices d'assurance pour financer ces coûts (dommage direct). L'impact du vol d'information (dommage indirect) étant beaucoup plus difficile à chiffrer et pas aussi clairement couvert avec un montant dans les **contrats d'assurance**.

Pendant que vous tentez de remédier à cette situation, rappelez-vous qu'il est souvent impensable d'arrêter les Systèmes d'Information pour une durée indéterminée. Il faudra donc assurer la vente pendant les travaux... et trouver des ressources !

En cas d'agression large où trouver cette réserve ? C'est toute la question des **réserves cyber** pour les états qui ont pris conscience du risque de **résilience** et d'agression numérique.

Les professionnels étant déjà occupés, il est impensable d'aller chercher des ressources en quantité sur le marché. S'appuyer sur une « **délégation de service public** » et utiliser des **étudiants** restent les viviers les plus probables.

Mettre en place une mécanique de réserve est complexe, et la maintenir avec des exercices l'est tout autant. La compétence informatique est de plus périssable et difficilement évaluable en masse. Comment peut-on mobiliser, organiser, coordonner ?

Ces questions commencent à trouver des éléments de réponses à travers la réserve opérationnelle, la réserve citoyenne de cyberdéfense et la réserve cyber à vocation opérationnelle. Autant d'instruments à construire ou à développer pour résoudre cette question primordiale : le *staffing*.

Gérer ces ressources est un défi permanent en gestion de crise.

En cas d'**infogérance** de tout ou partie du SI compromis, un élément de complexité vient se rajouter pour la gestion des ressources autour de la notion de MOA (Maîtrise d'Ouvrage). La force décisionnelle et contractuelle de la MOA permet une bonne gestion de crise.

Il est nécessaire alors de prévoir dans les contrats un cadre de pilotage en direct par la MOA pour employer des ressources supplémentaires, aménageant ainsi les obligations contractuelles de l'infogérant en tant normal. Typiquement il est nécessaire de stipuler dans le contrat d'infogérance qu'en cas de tentative, ou de compromission réussie, le client et les parties prenantes aidant le client dans cette crise, pourront bénéficier d'un accès complet au système d'information. En contrepartie, l'infogérant se verra décharger de tout ou partie de ces obligations et SLA pendant cette période et il sera soumis à un processus décisionnel différent qu'il conviendra d'anticiper dans le contrat.

Conclusion

J'espère que cet article aura permis d'entrevoir la complexité d'une gestion de crise d'une intrusion dans un grand parc. Il est nécessaire d'envisager très sérieusement ce scénario et de l'anticiper en s'exerçant, typiquement lors d'audit red team [5], qu'il ne faut pas hésiter à transformer en un exercice de crise.

Toutes les organisations se transforment et évoluent au contact d'une crise.

Une cybercrise n'est pas une fatalité si vous la transformez en opportunité. ■

■ RÉFÉRENCES

[1] <http://www.senat.fr/rap/r11-681/r11-6811.pdf>

[2] Voir les théories d'organisation d'Henry Mintzberg, en particulier « structure et dynamisme des organisations »

[3] <http://www.vmware.com/security/hardening-guides>

[4] <http://www.ssi.gouv.fr/fr/menu/actualites/appele-commentaires-referentiel-d-exigences-applicables-pris.html>

[5] http://en.wikipedia.org/wiki/Red_team

DÉBUSQUER L'APT DANS VOTRE PARC

Thomas Dumont – tdumont@silent.works



mots-clés : *INDICATEURS DE COMPROMISSION / AUDIT / LOGS SYSTÈME / APT / GPO / MALWARE*

Comment détecter un attaquant à l'intérieur de votre réseau ? Cet article couvre l'examen des points terminaux, c'est-à-dire les serveurs et les stations de travail.

Votre téléphone sonne, seul « numéro inconnu » s'affiche, il est 18h30 et vous savez que ce n'est pas bon signe. Bingo, c'est votre autorité de tutelle qui vous informe que l'un des ordinateurs de votre société se connecte au serveur d'un groupe d'attaquants qui était surveillé.

1 La réponse sur incident

1.1 Élaborer une campagne de recherche

Vous raccrochez, avec à la clef, quelques marqueurs qui vous permettront de détecter les malwares sur votre parc informatique. Ces fameux « indicateurs de compromission » (IOC) sont en fait des traces que laisse l'exécution d'un programme sur un système (on parle également d'artefacts).

Ces marqueurs peuvent être variés : empreinte MD5, nom de *named pipe*, nom de *mutex*, clé de registre, nom de fichier, extension particulière, empreinte de la table des imports (*imphash*), chaîne de caractères singulière, etc.

1.2 Exploitation des marqueurs

1.2.1 Préparation, conception

Fort de ces informations, il est maintenant temps de les exploiter. Le but de la réponse sur incident est d'identifier toutes les machines (serveurs, stations de travail ou équipements réseau) compromises. Pour

ce faire, il n'est pas envisageable de prendre chaque machine, dupliquer son disque et l'analyser avec des outils de forensic classique.

La première étape va être celle du *triage* : la présence d'IOC doit être vérifiée sur chaque ordinateur du périmètre supposé impacté. Classiquement, un agent sera déployé sur l'intégralité du parc. Deux approches complémentaires coexistent :

- collecter toutes les informations de la machine pour les transmettre à un serveur ;
- tester en direct la présence de marqueurs.

Ces deux approches sont très différentes au niveau de leur mise en œuvre. La détection locale est évidemment beaucoup plus simple à réaliser, mais reste problématique :

- les marqueurs vont potentiellement être dévoilés à l'attaquant, car il pourra récupérer votre code sur les machines qu'il maîtrise ;
- la détection sera limitée aux marqueurs que vous connaissez ;
- il vous faudra de toute manière un canal de communication pour obtenir la liste des machines infectées.

Mais la remontée de toutes les informations pertinentes des machines n'est pas sans écueils :

- le volume de données à transférer peut être important, de l'ordre de plusieurs dizaines voire centaines de Mo par machine ;
- les données collectées vont contenir des informations sensibles et personnelles, il faut donc assurer leur confidentialité ;
- le risque de données incomplètes est plus important ;
- l'accord de la victime quant au déploiement d'un agent aussi intrusif sera plus difficile à obtenir.



Les volumes attendus pourront également poser un défi aux équipes informatiques, car il faudra disposer :

- de plusieurs centaines de Go de stockage ;
- d'un serveur accessible par le réseau depuis toutes les machines et ceci, de manière sécurisée ;
- d'un moyen de communication apte à transmettre les volumes estimés.

1.2.2 Écriture de l'agent

Mais quel agent utiliser ? Si une liste au père Noël devait être écrite, elle commencerait ainsi (veuillez noter que dans l'intégralité de cet article, seul l'environnement Windows est traité) :

- autonome : un binaire unique qui ne dépend d'aucune bibliothèque et non adhérent à une version particulière de la machine virtuelle .NET ou à une version d'OS ;
- flexible : ce binaire doit être déployable sur tout le parc, en passant des serveurs (de production !) aux tablettes Windows 8. Aucune hypothèse de contexte d'exécution ne doit être prise ;
- transparent : la réussite de la campagne de recherche sera directement liée au taux de couverture du parc. Il est ainsi primordial que l'agent puisse tourner sur les postes de l'utilisateur sans impacter leur fonctionnement. Par expérience, un agent s'exécutant avec une priorité d'ordonnancement configurée à **BELOW_NORMAL_PRIORITY_CLASS** sera optimal. L'inconvénient est que son temps d'exécution sera rallongé, mais au final il importe peu que l'agent tourne en 10 ou 30 minutes ;
- robuste : même en cas de bugs, des mécanismes « ceinture et bretelle » devront être implémentés afin que l'exécution du programme soit *fail-safe*. Il est ainsi possible de mimer les *ulimits* d'Unix à l'aide des *Jobs Objects* et de limiter la consommation mémoire ou le temps d'exécution processeur de l'agent et de laisser le noyau tuer le processus en cas de dérive ;
- immune au *rootkits* (on a bien dit liste au père Noël !) ;
- faible pollueur : en police criminelle, il existe le principe d'« échange de Locard » qui stipule que lorsque deux corps entrent en contact l'un avec l'autre, il y a nécessairement un transfert entre ceux-ci. Il en est de même avec l'agent : il ne doit pas souiller la scène de crime. D'où un élément extrêmement important à retenir : pour chaque outil utilisé, il faut connaître son impact direct ou indirect. Un exemple d'impact direct : l'utilisation des outils SysInternals qui, par défaut, va écrire une clé de registre quant à l'acceptation de leur politique d'utilisation.

Un exemple d'impact indirect : la pollution du *prefetch* de Windows. Cette fonctionnalité, apparue avec Windows XP, met en cache de manière agressive les binaires les plus utilisés. Chaque fois qu'un binaire est exécuté, une entrée est ainsi ajoutée ou mise à jour dans le répertoire **%SystemRoot%\Prefetch**. Cependant, chaque entrée est une mine d'or pour l'analyse forensic puisqu'elle permet de connaître le nom de chaque exécutable lancé, la liste de ses DLL chargées, le nombre de fois que le binaire a été exécuté ainsi que l'horodatage de la dernière utilisation. Néanmoins, Windows limite la capacité de ce trésor d'information à seulement 128 entrées. Donc à chaque programme externe que votre outil de triage exécutera, vous perdrez une entrée du *prefetch*. Dommage !

Globalement, le seul moyen de garantir les contraintes énoncées ci-dessus est d'écrire l'agent en C/C++, ou *a minima* son cœur. L'utilisation d'un langage de plus haut niveau sera problématique à plusieurs égards :

- l'accès à l'API Windows devient une épreuve de contorsionnisme extrême ;
- la machine virtuelle .NET n'est pas installée par défaut sur Windows XP (mais vous n'avez plus un seul XP sur votre parc de toute façon, n'est-ce pas ?) ;
- déploiement alambiqué : par exemple le langage Python n'est pas le plus pratique sous Windows, car son déploiement nécessite bien trop souvent de créer une archive contenant le code source, l'interpréteur, ses DLL et tous les modules nécessaires, de l'envoyer sur le poste, de la décompresser puis d'exécuter le script. Autant dire que le principe de Locard vu précédemment va vous exploser au visage.

Un langage de haut niveau parfaitement adapté aux problématiques de déploiement est le langage Go de Google puisque son compilateur produit un exécutable complètement auto-porté et autonome. De plus, le code source est alors portable entre Linux, Windows ou Mac. Le seul hic reste la taille du binaire généré : autour des 3 Mo.

En conclusion, développer un agent autrement qu'en langage C/C++ est probablement un frein à long terme. La vérification de marqueurs simples sera en effet faisable très rapidement en Python, Powershell ou Visual Basic, mais les choses se compliqueront dramatiquement dès lors qu'il faudra toucher aux fonctionnalités avancées de Windows. Bien sûr, chaque langage fournit un moyen de FFI (*Foreign Function Interface*) qui rend la chose possible, mais « si le seul outil que vous avez est un marteau, vous tendez à voir tout problème comme un clou ».

Même en C, il y a des pièges : chaque nouvelle version de Windows apporte son lot de nouvelles bibliothèques, ou d'extensions de structures. Même si votre code compile, il n'est pas garanti qu'il s'exécute



correctement sur toutes les versions de Windows. Fort heureusement, tout cela est très bien documenté dans le MSDN.

Plus subtil, un autre piège est la chaîne de compilation utilisée. L'utilisation d'une version de Visual Studio trop récente (supérieure à VS2012) liera par défaut le binaire à une version d'API Windows qui ne sera pas disponible sur XP. Il est alors recommandé d'utiliser par exemple Visual Studio 2005 Express, qui est disponible gratuitement sur le site de Microsoft. Notez que vous n'aurez malheureusement pas accès aux nouvelles fonctionnalités de sécurité, ni à celles de C++11.

Le port de GCC sous Windows, MinGW, ne sera pas suffisant, car il ne supporte pas la macro Windows UNICODE. Or, certaines fonctions de Windows (comme la vérification de signature Authenticode) sont uniquement disponibles en version Unicode.

Si vous n'avez pas le temps, les ressources ou l'énergie de développer votre propre agent, rassurez-vous, il existe aujourd'hui une multitude d'agents disponibles : libres, gratuits ou propriétaires. Parmi eux, citons-en quelques-uns : Mir, Grr, CrowdResponse, Kansa, Kraken ou FastResponder.

1.3 Déploiement de l'agent

Un facteur critique à la réussite de l'opération de collecte ou de recherche est la couverture du parc. En effet, si notre agent ne s'exécute que sur 30 % des machines, le risque de ne pas trouver des machines infectées est élevé.

Il faut donc mettre en œuvre une méthode de déploiement fiable qui réponde aux besoins suivants :

- exécution périodique ;
- exécution en contexte SYSTEM, pour disposer des droits nécessaires à la collecte des éléments de bas niveau du système ;
- exécution en contexte utilisateur, pour la collecte des éléments de persistance ou de compromission liés à un compte utilisateur ;
- remontée des résultats sur une ressource authentifiée ;
- fonctionnement sur tout le parc, y compris les éventuelles machines hors domaine.

Évidemment, il va être difficile de trouver une méthode qui remplisse tous ces critères. De plus, les administrateurs du parc sont souvent réticents à déployer un binaire sans connaître les conséquences possibles.

On préférera donc en priorité les techniques déjà utilisées par les administrateurs lorsqu'elles existent, par exemple d'éventuels outils de gestion de parc tels que SCCM, ZENworks, LANPark et SCOM dans une moindre mesure.

Si de tels outils ne sont pas disponibles, on privilégiera les techniques standards de Microsoft :

- GPO *startup* machine pour une exécution au démarrage de la machine dans un contexte d'exécution SYSTEM ;
- GPO *user* pour une exécution à l'ouverture de la session d'un utilisateur.

Cependant, l'utilisation de base des GPO pose plusieurs problèmes :

- les postes qui ne redémarrent pas ne sont pas couverts ;
- les utilisateurs qui ne s'authentifient pas non plus ;
- les GPO sont normalement restreintes en temps d'exécution ;
- elles peuvent être configurées en mode synchrone où le poste est bloqué jusqu'à la fin de l'exécution de tous les scripts, empêchant la connexion de l'utilisateur.

Attention !

Certaines de ces limitations peuvent être contournées utilisant les *jobs* ou en lançant l'agent via un script WMI.

Les serveurs posent donc particulièrement problème, car l'on ne va généralement pas les redémarrer pour exécuter une GPO. De même, les postes hors domaine ne peuvent évidemment pas être atteints ainsi.

Cependant, depuis Windows Vista et 2008, Microsoft a ajouté des fonctionnalités aux GPO baptisées « GPO de préférence ». Parmi ces fonctionnalités, on trouve les tâches planifiées à exécution immédiate qui offrent la possibilité d'exécuter un programme sans redémarrage ni ouverture de session d'un utilisateur. Cette solution peut donc être envisagée sur un parc plutôt récent.

Si les GPO ne sont pas utilisées, il reste des techniques beaucoup moins propres :

- *PSEXec* en masse, mais qui laisse des traces similaires à une activité malveillante ;
- les tâches planifiées, définies à distance avec l'outil **at.exe** ou **schtasks.exe**. Cette technique fonctionne bien, mais il est impossible d'obtenir un retour d'erreur simplement et son utilisation par les attaquants peut également laisser penser à une activité malveillante ;
- WMI (via la classe **Win32_Process**) qui permet d'exécuter des commandes à distance. Mais comme l'authentification pour l'exécution du script est réalisée par réseau, le script n'a pas la possibilité d'utiliser l'authentification implicite pour la dépose des résultats ;



- WinRM/WinRS est une technique native et propre, mais intégrée uniquement depuis Vista/2008 et ne permet donc pas de couvrir un vieux parc ;
- en dernier recours, on pourra utiliser un script **xdotool** pour injecter des frappes clavier dans une session graphique de type RDP ou VNC.

Il faut noter que beaucoup de ces solutions ne fonctionnent pas pour des systèmes hors domaine. En effet, à partir de Vista, le mécanisme UAC restreint les comptes locaux et ceux-ci perdent leur droit d'administration lorsqu'ils s'authentifient par le réseau.

Une fois le déploiement effectif, il faut mesurer le taux de couverture du parc. Si celui-ci est trop faible ou ne progresse pas assez vite, une solution consiste à demander explicitement le redémarrage des machines clientes, sous couvert, par exemple, d'opérations de maintenance.

Ainsi s'achève la section sur l'exploitation de marqueurs préalablement découverts (par une source tierce, le reverse engineering d'un malware ou le résultat d'analyse dynamique par une sandbox telle que Cuckoo). La section suivante s'attachera à découvrir l'inconnu, tout un programme !

2

Audit d'un parc informatique

Tout part d'une seule information : des documents métier ont été découverts sur le serveur de C&C d'un groupe d'attaquants. Comment remonter aux machines compromises par l'attaquant et à ses malwares ?

Alors que dans la section précédente, on recherchait des marqueurs connus sur le parc, ici, nous allons devoir utiliser des heuristiques pour trouver les traces malveillantes.

Quels sont les artefacts que l'on peut exploiter ? Probablement :

- un moyen de persistance est nécessaire pour exécuter le malware à chaque redémarrage ;
- une communication vers Internet ;
- l'utilisation d'un compte à hauts privilèges ;
- la latéralisation de l'attaquant.

La méthodologie est alors la suivante : à partir de la collecte de toutes les informations systèmes de chaque ordinateur du parc de l'entreprise (station de travail ou serveur), la mise en évidence d'anomalie doit permettre *in fine* de découvrir un logiciel malveillant.

2.1 Méthodologie

Une fois que toutes les informations ont été collectées sur tout le parc, vous avez en votre possession, si tout va bien, un inventaire complet de :

- tous les processus persistants ;
- toutes les empreintes des exécutables présents sur les disques durs ;
- toutes les connexions réseau en cours (en écoute ou connectées) ;
- tous les processus en cours d'exécution.

C'est une véritable mine d'or. Si en plus, vous avez fait la même opération sur l'image « maître » utilisée pour installer tous les systèmes des postes de travail, vous avez la capacité de faire une liste blanche réduisant ainsi la quantité d'information à traiter.

La clef de l'analyse est cette réduction d'information. Par exemple, dans un premier temps, il est intéressant de filtrer tout item présent sur plus de 90% du parc (dans le cadre d'un parc de plusieurs milliers de machines), car si jamais l'attaquant a compromis autant de machines, fermez ces pages et brûlez tout.

Généralement, l'audit se passe en trois étapes :

- analyse (via une heuristique) ;
- récupération des fichiers suspects sur les postes ;
- triage (reverse engineering, réputation du binaire).

En pratique, chaque tâche occupe au moins une personne à plein temps, il est donc recommandé que ces personnes travaillent en étroite collaboration, si possible dans la même salle pour booster la réactivité.

Dans la suite de cette section, nous nous attacherons uniquement sur la tâche d'analyse.

2.1.1 Low hanging fruits

Commençons par les fruits prêts à tomber sur le sol :

- tous les processus tournant avec les droits super-utilisateurs ;
- tous les processus avec un moyen de persistance ;
- tous les processus utilisant le réseau ;
- tous les processus non-signés ;
- tous les processus dont l'exécutable se trouve dans un répertoire temporaire ou système.

Si une liste blanche est utilisée, le travail pourra être rapide. Mais il ne faut pas oublier que des binaires légitimes, voire signés par Microsoft, peuvent être utilisés à des fins malveillantes !

2.1.2 Anomalie statistique

Tout exécutable, service ou processus présent sur moins d'un pourcent du parc est suspecté et doit être vérifié. Suivant la localisation de ce binaire, comme dans le répertoire utilisateur par exemple, il est rapide de se faire une idée sur l'innocuité.



Ensuite, il faut également combiner toutes les autres informations collectées : si cet exécutable utilise le réseau et qu'un moyen de persistance est constaté alors que le binaire est totalement inconnu, c'est louche.

Mais évidemment, tout n'est pas si simple et souvent ces binaires seront légitimes, par exemple un *driver* pour du matériel spécifique, non signé par l'éditeur, qui possède 100 versions différentes.

2.1.3 Noms dérivés

Les attaquants exploitent à leur avantage les faiblesses de l'être humain avec la dérivation de noms légitimes. Il suffit ainsi d'une faute d'orthographe ou d'une faute de typographie volontaire pour que le cerveau de l'être humain soit berné et corrige la faute de manière transparente.

C'est pour cette raison que les auteurs de malwares aiment bien nommer leurs exécutables avec des noms d'anti-virus, car ils ont souvent plusieurs syllabes faciles à maquiller.

D'ailleurs, si l'entreprise que vous auditez utilise l'anti-virus Symantec, il sera malin de vérifier l'absence de tous les autres anti-virus du marché.

Certains installent leurs exécutables dans des dossiers de services existant au préalable, tout en les nommant de manière plausible, ce qui rend la détection par le nom nettement plus difficile.

2.2 Persistance

Une fois qu'un attaquant a posé et exécuté son RAT (outil communiquant avec le serveur de contrôle), il doit s'assurer que celui-ci survivra aux redémarrages de la machine.

Une encyclopédie de tous les moyens de persistance serait difficilement exhaustive et plutôt ennuyeuse. Cependant, les moyens les plus populaires sont les suivants :

- création d'un service Windows ou installation d'un driver. C'est le moyen le plus sûr puisque c'est exactement le rôle de cette fonction de Windows : faire tourner des codes en tâche de fond dès le démarrage d'un système. Classiquement, les attaquants utilisant ce vecteur prennent le nom d'un service ou d'un driver ressemblant à quelque chose de légitime ;
- base de registre : c'est une pépite d'or pour les attaquants vu le nombre de clés où il est possible d'assurer sa persistance tout en étant plus ou moins dissimulé. L'outil **autoruncs** de SysInternals est un des logiciels le plus complets pour lister toutes les

clés à disposition de l'attaquant. Une page web tente de tenir à jour cette liste d'autorun sur <http://gladiator-antivirus.com/forum/index.php?showtopic=24610> ;

- DLL Injection (**Appinit_Dlls**) : Avant Windows Vista, tout exécutable lié à **User32.dll** chargeait automatiquement toutes les DLL listées dans la clef de registre **Appinit_Dlls**. Cette fonctionnalité a été désactivée par Windows Vista, puis est réapparue dans Windows 7 avec le support de la signature de code qui est désormais obligatoire. Néanmoins, étant donné que les groupes d'attaquant ont déjà à leur disposition des montagnes de certificats volés pour signer leur binaire, l'apport en sécurité est quasi nul ;
- *DLL Search Order hijacking* : Lorsqu'une application Windows charge une bibliothèque via LoadLibrary, l'OS va rechercher cette DLL à partir de chemins prédéfinis : le premier choix est le répertoire de l'exécutable, puis d'autres positions. Une exception notable est le cas d'une clef de registre, **KnownDLLs** : si le nom de la DLL à charger est présent, alors elle sera forcément chargée depuis **c:/windows/system32**. Un moyen de persistance très discret est alors de placer une DLL malveillante dans le répertoire courant d'une application légitime afin d'être chargée automatiquement (mais cette DLL à surcharger ne doit pas être contenue dans *KnownDLLs*).

2.3 Chercher la trousse à outils

Si vous n'arrivez pas à détecter le RAT de l'ennemi, il reste toujours la possibilité de le pister grâce à sa trousse à outils et ses artefacts. Elle se compose la plupart du temps de :

- un archiveur, RAR étant le plus populaire ;
- un extracteur de base Active Directory (QuarksDump, Windows Credential Editor, PWDump) ;
- un outil de scan NetBios.

En plus d'implémenter des règles pour détecter ces binaires ainsi que leurs artefacts, il faut également pouvoir rechercher le résultat de la sortie de ces outils.

Dans certains cas, il est possible de voir l'attaquant développer son outil au fur et à mesure de l'attaque. Il n'est pas rare ainsi de trouver une version de binaire différente entre chaque machine, ce qui est plutôt intéressant à analyser puisqu'on découvre les besoins de l'attaquant via l'ajout de fonctionnalités.

Malheureusement, l'utilisation de versions différentes implique qu'il n'existe pas d'empreinte MD5 unique pour détecter toutes ces versions dérivées. Au mieux, il faudrait utiliser du « *fuzzy matching* » comme l'outil **ssdeep** ou **sdhash** qui, pour deux binaires légèrement



différents, va retourner deux empreintes très proches. En pratique, il n'est pas réaliste de calculer ces signatures de cette façon dans un audit, car cette méthode ne passe pas à l'échelle.

FireEye Mandiant a alors imaginé d'utiliser une particularité des fichiers PE : l'ordre d'import des bibliothèques, qui est fixé par le développeur. Ils se sont rendu compte que cet ordre était consistant dans le temps, malgré les versions successives. Un *hash*, appelé *imphash*, des imports concaténés dans l'ordre peut alors éventuellement caractériser un binaire.

3 Les logs systèmes

Intéressons-nous maintenant à la latéralisation de l'attaquant sur un parc informatique, autrement dit, ses techniques pour se déplacer d'une machine à une autre, principalement par :

- exécution de tâches planifiée ;
- connexion interactive (RDP ou PSEXEC) ;
- montage de disque dur (commande **net use**).

La détection de ces activités, et bien d'autres, passe par l'analyse des journaux systèmes de Windows dont les trois principaux sont *Application*, *System* et *Security*. Lorsque ceux-ci sont bien configurés, la quantité d'information utile qui peut en être extraite est inestimable. Mais la triste vérité, c'est que la configuration par défaut sous Windows XP et 2003 est médiocre et largement perfectible depuis Windows Vista et 2008. En effet, par défaut, Windows ne journalise pas tous les événements de sécurité et la taille des journaux, en particulier celle du journal *Security*, est souvent trop faible. Sur les serveurs générant beaucoup d'événements, en particulier sur les contrôleurs de domaine, la durée de rétention des événements n'est pas suffisante pour être exploitable (« 3 jours de conservation is enough »).

Cependant, avant de modifier les politiques de journalisation de Windows et afin d'éviter de perdre des entrées, il est indispensable de récupérer au préalable les journaux sur tous les systèmes. En effet, tout changement de la politique de journalisation risque de générer un volume plus important d'événements et ainsi de remplir plus rapidement des journaux. Or, dans la majorité des cas, ceux-ci sont circulaires, c'est-à-dire que les anciens événements sont supprimés s'il n'y a plus de place.

Après cette phase de séquestre initiale, il est possible d'améliorer la politique de journalisation tout en en profitant pour augmenter la taille des journaux sur tous les systèmes. Il faut cependant savoir que, sous Windows XP et 2003 en 32 bits, la taille totale des journaux est limitée à 300 Mo.

Un événement Windows se caractérise par son fournisseur (*provider*) et son numéro d'événement (*EventId*). Le fournisseur le plus important du point de vue de la sécurité est *Microsoft-Windows-Security-Auditing* qui a remplacé le fournisseur *Security* des systèmes Windows XP et 2003 (un décalage de 4096 sur les *EventId* existant pour les événements similaires). Ces deux fournisseurs écrivent leurs événements dans le journal *Security* qui, contrairement aux deux autres journaux, nécessite des droits d'administration pour être consulté.

Plusieurs catégories d'événements et scénarios doivent être recherchées dans les journaux. En premier lieu, il y a celles relatives à l'authentification qui permettent de suivre l'activité d'un compte utilisé par un attaquant. Pour cela, l'événement le plus important est le 4624 (528 et 540 sous Windows XP/2003) généré pour chaque ouverture de session (interactive, réseau, service, etc.) suite à une authentification réussie.

Il est également possible, sur les contrôleurs de domaine, de regarder la validation des authentifications NTLM (4776) ou la délivrance de ticket Kerberos (4768 et 4769). Cependant, cette méthode est très limitée. D'une part l'événement 4776 ne journalise pas quel système était destinataire de l'authentification. D'autre part, la conception du protocole Kerberos ne permet pas d'être pertinent en analysant uniquement la délivrance des tickets Kerberos. En effet, les contrôleurs de domaine ne journalisent que l'émission des tickets, mais pas leur utilisation. Or un ticket peut servir à s'authentifier autant de fois que nécessaire, et ce, pendant toute sa durée de validité qui est, par défaut, d'une heure pour un ticket de service.

Ainsi, se limiter aux journaux des contrôleurs de domaine ne permet que de voir l'émission d'un ticket de service alors que l'attaquant aura pu de connecter des centaines de fois au serveur distant, ou pas du tout. De même, toutes les attaques consistant à forger des tickets ne seront pas détectées. Pour avoir un suivi pertinent des authentifications, la collecte des événements sur le parc entier est indispensable pour ne rien manquer et se limiter aux seuls contrôleurs de domaine est très insuffisant.

Outre les événements liés à l'authentification, d'autres catégories d'événements doivent être analysées. Concernant les services, souvent utilisés par les attaquants pour élever leurs privilèges au niveau SYSTEM ou exécuter des commandes à distance (PSEXEC), l'analyse de l'activité services passe par l'inventaire des événements de notification à un service (Service Control Manager/7035), de changement d'état (Service Control Manager/7036) et d'installation (Service Control Manager/7045 et Microsoft-Windows-Security-Auditing/4697).

Les tâches planifiées doivent également être surveillées : création (4698) et démarrage (Microsoft-Windows-TaskScheduler/100). À noter que sur les systèmes Windows XP/2003, beaucoup d'actions du gestionnaire de tâche sont également enregistrées dans le fichier **SCHEDLGU.TXT**.

Connaître les programmes exécutés sur un système se révèle aussi extrêmement utile et plusieurs sources d'événements peuvent être utilisées. Tout d'abord, si le suivi de processus est activé, celui-ci génère un événement 4688 pour chaque exécution de processus. De même, si une stratégie de restriction logicielle est activée (SRP ou AppLocker), de nombreux événements sont générés :

- Microsoft-Windows-AppLocker/8002 et 8004 : autorisation/refus d'exécution d'un EXE ;
- Microsoft-Windows-AppLocker/8005 et 8007 : autorisation/refus d'exécution d'un script ;
- Software Restriction Policies/865 à 866 : blocage d'exécution d'un programme.

Enfin, si le service Identité de l'application (AppIDSvc), en charge de la vérification des signatures numériques des exécutables, est en cours d'exécution, celui-ci reportera ses vérifications via l'événement Microsoft-Windows-AppID/4004.

Cette liste d'événements est très succincte et il existe bien d'autres événements qui permettent de détecter une activité malveillante : défaillance d'application, *crash* du noyau, verrouillage de compte, chargement de composant de sécurité (AP, SSP) dans LSASS, etc.

Pour pouvoir analyser les journaux Windows, il faut préalablement les centraliser. Avec Windows XP/2003, la collecte des journaux Windows via le réseau était complexe et rendait indispensable l'utilisation d'un logiciel tiers. Mais Microsoft s'est depuis fait pardonner grâce à *Windows Event Collector* (WEC), une solution basée sur WinRM qui permet nativement de collecter et de centraliser les journaux. Depuis Windows 2008, il est ainsi possible de sélectionner finement les événements à remonter à un collecteur et de définir une stratégie d'utilisation de la bande passante : privilégier une faible latence et envoyer les événements en direct ou bien passer en mode « par salve » pour ne pas saturer le réseau en permanence.

Conclusion

Cet article n'a fait qu'effleurer certains des aspects du traitement d'incident sur un grand parc de machines. Ce domaine né récemment en est encore à ses balbutiements et devra nécessairement progresser pour faire face aux codes malveillants les plus avancés. Imaginez-vous devoir rechercher les traces d'une compromission sans IOC, sur un parc hétérogène Windows/Linux et où l'attaquant utilise des *rootkits* et des implants de type *IRATEMONK*... ■

NE LE MANQUEZ PAS : OPEN SILICIUM N°12 !

FOCUS : PRENEZ EN MAIN LES TECHNOLOGIES RFID / NFC



**DISPONIBLE CHEZ
VOTRE MARCHAND
DE JOURNAUX ET SUR :
boutique.ed-diamond.com**



INTRODUCTION AU TRAITEMENT DES LOGS ET DES FLUX RÉSEAU DANS UNE RÉPONSE SUR INCIDENT

Guillaume Lambert – glambert@silent.works

mots-clés : LOGS / TRACES RÉSEAU / RÉPONSE SUR INCIDENT / APT / MÉTADONNÉES / SONDES

Cet article présente une introduction à l'analyse des logs et des flux réseau dans le contexte d'un incident, de type APT ou SPT (Simple Persistent Threat pour les intimes), sur un grand parc client. Ce domaine étant vaste, cet article d'introduction s'attachera à comprendre et analyser les besoins et les aspects pratiques de l'analyse des logs et du réseau. Il se poursuivra sur des exemples, idées et axes de réflexion utiles sur le terrain d'une réponse sur incident.

1 Le passé de l'incident - Traitement des logs

1.1 Besoin, état de l'art

Le traitement des logs est l'exercice d'analyse réalisé, le plus souvent, au cours d'une réponse sur incident pour évaluer rapidement la gravité d'une compromission survenue dans le passé. Pour des aspects légaux [1] et de simplicité de manipulation, l'étude des logs est souvent préférée à une analyse des flux réseau, laquelle nécessite des modifications et des interventions sur l'architecture réseau. L'analyse des flux réseau sera, elle, utilisée pour une analyse plus approfondie et la réactivité au fil de l'eau.

Dans le cadre des incidents définis en introduction, l'analyse des logs sera le plus souvent l'étude des journaux proxy. On partira du principe que ces derniers sont effectivement collectés et archivés de façon centralisée et ont pour utilité :

- identifier les machines et comptes compromis ;

Note

Le terme « compte », dans cet article, fait référence au compte d'authentification utilisé sur les proxy, VPN, etc.

- calculer les volumétries montantes et descendantes vers chaque serveur C&C, ce qui peut se révéler utile pour estimer la quantité de données potentiellement exfiltrée et la taille des messages de *keep alive* ;

- identifier un maximum de domaines malveillants selon un ou plusieurs *indicator of compromise* (IOC) ;

Pour cela, une bonne analyse de logs nécessite au préalable de :

- uniformiser les formats selon les types de journaux (applications, équipements, etc.) ;
- collecter de manière régulière, quelle que soit l'hétérogénéité du parc ;
- centraliser, archiver et chiffrer. Notons que les séquestres de logs, qui garantissent leur intégrité, seront un atout non négligeable ;



- surveiller le bon fonctionnement de ces derniers points, afin de s'assurer qu'aucun champ ou fichier de logs ne soit manquant.

Note

Les guides et recommandations de l'ANSSI doivent être respectés pour parvenir à une bonne mise en œuvre d'un système de journalisation [2] [3].

Bien que, la majorité du traitement des logs soit focalisée sur les journaux proxy, plusieurs types de logs sont à la disposition de l'analyste pour rechercher un maximum d'éléments liés à la compromission.

1.2 En pratique

Pour parvenir à cela, l'analyse des logs sur le terrain doit faire face aux exigences suivantes :

- gestion d'une grosse quantité de données, l'incident pouvant durer depuis plusieurs mois ou années ;
- recherches performantes, pour comprendre rapidement l'incident ;
- recherches adaptatives et évolutives.

1.2.1 Quantité de données

La quantité de données dans l'analyse de logs est une contrainte importante à prendre en compte. En effet, certains types de journaux peuvent facilement atteindre plusieurs Go par jour et il est fréquent d'avoir à analyser plusieurs To de fichiers de logs compressés, pour une étude portant sur plusieurs mois. Dans ce contexte, des actions triviales telles que copier ou décompresser, deviennent problématiques. Les technologies de stockage et les types de supports employés (disque dur SSD, USB, SAS, etc.) doivent être adaptés à la situation. Prendre un petit peu de recul avant de commencer l'analyse ou l'indexation sera primordial pour éviter de perdre un temps précieux dans la réponse sur incident.

Ainsi, il sera souvent important de suivre les règles suivantes :

- réduire la quantité de logs. L'utilisation de *whitelists* (sites connus, internes, partenaires, etc.) est le plus souvent effectuée pour réduire la quantité de logs, mais cette opération peut fausser l'analyse (pivot de sites internes ou partenaires, compromission d'un site légitime, etc.). Pour y remédier, une analyse des logs évincés par la *whitelist* sera possible dans un second temps ;
- compresser : si possible compresser les données pour diminuer la taille et améliorer les performances ;

- éviter les intermédiaires : connecter le support de données au plus proche du système d'analyse. Éviter les copies au travers du réseau et connecter le disque directement sur votre système. Dans certains cas, l'envoi d'un disque dur, par la Poste, est plus rapide que le transfert via un VPN, même si on compte le temps pour chiffrer les données sur le disque et l'expédition de celui-ci. Dans le cas de gros fichiers, il sera toujours préférable de commencer à analyser les logs directement sur le support fourni, au lieu de faire la copie sur le système d'analyse et ensuite lancer l'indexation/analyse.

1.2.2 Recherches performantes

Pour parvenir à optimiser les recherches, l'analyste doit comprendre les goulots d'étranglement et les minimiser.

Pour des traitements de recherche de motifs ou statistiques légers, tels que décrit dans cet article, les entrées-sorties sont généralement le facteur limitant. Rien ne sert d'investir dans des machines contenant des centaines de cœurs et To de RAM. En effet, le traitement par le CPU ou la place utilisée dans la RAM seront négligeables devant l'attente des informations lues sur le disque. Pour cette raison, il faut éviter de lire plusieurs fois la même donnée depuis le disque dur. Ainsi, il faudra utiliser de préférence des logs compressés pour maximiser la quantité d'informations utiles récupérées par chaque accès au support et d'autre part utiliser un moteur d'indexation pour minimiser le nombre d'accès aux fichiers.

Il existe de nombreuses solutions d'indexation et de recherche pour le traitement en masse, comme Splunk ou RSA SA. Toutefois il est également possible de bricoler, pour des analyses ponctuelles, avec la solution libre ELKS (*Elasticsearch Logstash Kibana Stack*) qui est disponible pour Windows et Linux. Le guide « Setting up a single ELK node in 20 minutes » [4] est conseillé pour débuter ELKS.

1.2.3 Recherches adaptatives et évolutives

L'analyste sous pression, qui se doit d'être efficace en période d'incident, doit déterminer quels sont les métadonnées ou champs les plus intéressants à étudier et à indexer. Foncer tête baissée dans l'indexation massive des données brutes est contre-productif. Il préférera au préalable définir l'ensemble des champs importants selon les types de logs et les trier en fonction de leur pertinence. Celle-ci est définie par la fréquence d'utilisation de la métadonnée dans les IOC, l'importance du champ, son utilisation dans plusieurs



types de logs (qui pourra alors servir de valeur pivot) et l'expérience de l'analyste. En pratique, on retrouvera les champs suivants selon les types de journaux.

Proxy et Firewall :

- FQDN, domaine racine et adresse IP distante pour chaque requête ;
- volumétries descendantes et montantes par FQDN, domaine racine, IP et compte utilisé ;
- nombre de *hits* par FQDN, domaines racine et IP ;
- *path* et *query* de l'URI [5] pour chaque requête, FQDN, domaine racine et IP ;
- *user agent* utilisé par chaque FQDN, domaine racine et IP. L'apparition d'une valeur unique ou non standard peut identifier la présence d'un malware dans certains cas (*user agent* inscrit en dur dans le code du binaire, générés à chaque exécution du malware ou contenant des informations pour l'interaction avec le serveur C&C) ;
- comptes utilisés pour chaque FQDN, domaine racine et IP : utile pour détecter les comptes compromis. Cette métrique sera aussi très utile lors de l'identification d'un compte compromis, pour connaître l'activité réseau, détecter rapidement de nouveaux ou anciens C&C et étudier son importance dans l'attaque ;
- comptes utilisés par IP source : intéressant pour détecter l'utilisation de plusieurs comptes sur une machine donnée. Des faux positifs peuvent apparaître sur les machines partagées. Cette information sera aussi utile pour détecter les comptes utilisés par des malwares pour réaliser l'authentification sur les proxy ;
- adresse IP source pour chaque FQDN, domaine racine et IP distante : cette métadonnée identifie l'ensemble des machines compromises. Elle permet également de découvrir des nouvelles ou anciennes connexions malveillantes et d'étudier l'importance d'une machine compromise dans l'attaque.

Note

Une attention particulière doit être apportée pour correctement identifier la machine selon l'adresse IP et son nom et ses différents rôles dans l'attaque. Les logs DHCP et VPN seront indispensables pour assurer l'identification des machines utilisateurs.

Serveurs de messagerie (protocoles IMAP, SMTP, Exchange, etc.) :

- couple des champs From et To pour chaque e-mail ;
- volumétries descendantes et montantes totales échangées pour chaque couple From et To ;

- nombre de hits pour chaque couple From et To ;
- sujet de l'e-mail pour chaque couple From et To ;
- volumétries descendantes et montantes totales échangées pour chaque e-mail contenant un sujet donné ;
- nombre d'e-mails ayant le même sujet ;
- noms et tailles des pièces jointes pour chaque e-mail.

2 Le présent de l'incident - Le flux réseau

2.1 Besoin, état de l'art

L'analyse d'un flux réseau fera suite, lorsque l'attaque est toujours en cours, à l'analyse de logs classique décrite précédemment. L'analyste habile, qui possède désormais une meilleure connaissance de l'attaque en cours pourra alors suivre l'attaquant au jour le jour et détecter, pendant et après la remédiation, les postes encore compromis et les tentatives de réinfection. Pour cela, la solution idéale d'aide à l'analyse des flux réseau doit prendre en compte les exigences suivantes :

- réactivité et surveillance. La détection automatique et en temps réel fournit une solution d'alerte efficace pour observer les mouvements de l'attaquant ;
- flexibilité des recherches et de la détection. La solution doit permettre de mettre à jour les recherches et règles sur l'ensemble des champs et du contenu des flux réseau (calcul des volumétries, notification selon des seuils, détection d'incohérences protocolaires et de patterns spécifiques) ;
- enregistrement des métadonnées et du trafic. La collecte et la sauvegarde des métadonnées récupérées sur le flux réseau sont indispensables pour réaliser, de manière rétrospective, la découverte de nouveaux marqueurs, ou pour détecter la présence d'un nouveau marqueur sur l'ensemble du trafic examiné. Dans certains cas, l'enregistrement de différents types de trafic est souhaité (notamment le trafic depuis et à destination des serveurs C&C). On notera que les solutions de type *Full Packet Capture* (FPC) demandent des équipements spécifiques, relativement coûteux, lorsque le trafic observé est supérieur à 10 Mbps.

Bien qu'il existe plusieurs équipements réalisant l'analyse de flux réseau, peu offrent la flexibilité et les besoins que nécessite ce type d'analyse dans le cadre d'une réponse sur incident. En effet, chaque incident apportera de nouvelles problématiques et l'analyste dégoûdi souhaitera pouvoir modifier, contrôler et



faire évoluer sa solution. Ainsi, il n'est pas rare qu'il réalise sa propre sonde basée sur plusieurs briques existantes ou maison, s'appuyant sur des NIDS/NIPS, comme Snort et Suricata, des FPC (OpenFPC) ou des outils classiques, comme **tcpdump**, **ngrep**, **faup**, etc.

2.2 En pratique

2.2.1 Placement des sondes

La topologie et la complexité des réseaux de grande entreprise demandent à l'analyste clairvoyant de définir de manière fine le placement des sondes, qui peut varier en fonction de ce qu'il souhaite étudier. La première étape de cette étude consiste à choisir entre le traitement des sorties Internet de navigation Web et/ou de messagerie. En effet, les grandes entreprises décident fréquemment, pour de multiples raisons de performances, de SLA ou encore de contraintes opérateur, de séparer ces accès. Dans le cas d'une analyse de la sortie de navigation Web, l'analyste pourra choisir parmi les configurations suivantes :

- entre Internet et le proxy : ce positionnement frontal est souvent préféré pour analyser l'ensemble du trafic entrant et sortant. Cependant, il ne permet pas de connaître la source des requêtes Web, ni le compte utilisé. Les tentatives échouées (bloquées, non authentifiées, etc.) ne seront pas analysées par la sonde ;
- derrière le proxy ou au milieu. Pour résoudre les problèmes évoqués ci-dessus, le placement de la sonde derrière le proxy ou au milieu d'une chaîne de proxy peut être envisagé. Ainsi, la sonde aura une vision de toutes les demandes de connexions effectuées par le LAN, les requêtes pouvant être bloquées par un ou plusieurs équipements après la sonde. Dans le cas d'un positionnement au milieu d'une chaîne de proxy, le champ **X-Forwarded-For** devra impérativement être journalisé pour déterminer la source originelle de la requête ;
- dans le LAN et/ou la DMZ. Pour des cas spécifiques, il est parfois utile de se placer devant une zone spécifique (comme une DMZ) ou encore directement sur le LAN. Généralement, cette position détectera les machines internes servant de pivot ou proxy pour l'attaquant, surveillera très finement et au fil de l'eau les accès à un système compromis, ou encore étudiera les accès aux partages réseau.

En pratique, l'analyste sagace trouvera un compromis entre ce qu'il souhaite surveiller, les différentes positions possibles qui s'offrent à lui et le type d'équipement dont il dispose. L'utilisation de filtres sera un bon compromis pour traiter par exemple uniquement le port 445 et analyser les accès SMB.

2.2.2 Récupération ou duplication du trafic

Récupération du trafic

La récupération des flux réseau est souvent réalisée par un équipement (switch, routeur, etc.) au travers d'un *port mirroring*. Il consiste à copier le trafic circulant sur un ensemble de ports donnés vers un port unique. Cette méthode est souvent préférée par les administrateurs et les opérationnels du réseau pour des raisons de simplicité. Elle ne requiert aucun ajout d'équipement dans le réseau, ni interruption de service et la configuration leur est rapide et maîtrisée. Cependant, cette solution comporte deux limitations :

- le groupement du nombre de ports. La configuration d'un *port mirroring* dans certains équipements limite le nombre de ports utilisables. Dans certains cas, le groupement de plusieurs ports n'est pas possible, ce qui entraîne l'utilisation d'un port supplémentaire pour chaque port que l'on souhaite mettre sous surveillance ;
- la possible perte de trafic. Pour des raisons de performances, le *port mirroring* n'est pas une action prioritaire dans le fonctionnement d'un équipement réseau. Ainsi, la copie de l'intégralité du trafic peut ne pas être garantie et des pertes de paquets IP peuvent survenir lorsque la charge de l'équipement réseau est trop élevée.

Duplication du trafic

La duplication du trafic demande l'ajout d'un équipement spécifique dans le réseau. Le plus souvent, un boîtier TAP (*Test Access Port*) est utilisé, mais dans certains cas, un *hub* peut convenir. Ceux-ci se trouveront généralement au fond de la pièce de stockage de matériel ou dans les caves, au milieu d'une pile de technologies et équipements obsolètes. L'utilisation de TAP garantit que l'ensemble du trafic est correctement dupliqué, mais comporte les limitations suivantes :

- le groupement de ports. Sans ajouter un switch TAP matriciel ou en utilisant certains modèles de TAP, le groupement de ports n'est pas possible ;
- l'agrégation du trafic en émission et en réception. Certains modèles de TAP n'offrent pas la possibilité d'agréger ces deux trafics, ce qui entraîne la récupération du trafic dupliqué sur deux ports. Dans ce cas, la sonde aura la charge de reconstruire les flux.

Sur le terrain d'une réponse sur incident, le *port mirroring* sera préféré afin de fournir une solution rapide et réversible. Il permettra de récupérer plusieurs flux simultanément en minimisant les interventions des administrateurs, ce qui est très important pour garder une bonne entente entre les équipes dans cette période de stress intense.



2.3 Traitement des pcaps

Le bon traitement des résultats des captures réseau, ou pcaps, par la sonde est indispensable pour extraire correctement les informations utiles et il sera important de traiter de manière efficace les aspects suivants :

- la gestion automatique des pcaps. Pour garantir le bon fonctionnement de la sonde et l'extraction des informations utiles à l'analyse, les fichiers pcaps doivent être sauvegardés de manière tournante, compressés et supprimés après leur date d'expiration ;
- la reconstruction des informations applicatives. Ce problème demande un moyen d'extraire, automatiquement et au fil de l'eau, les informations échangées au niveau applicatif. À l'exception de certains encodages ou structures spécifiques au niveau applicatif (ASN.1, MIME, etc.) la principale difficulté de cette action provient de la reconstruction et la reproduction des mécanismes réseau. Reproduire le fonctionnement d'une pile TCP/IP ou le *HTTP pipelining* n'est pas un exercice facile, surtout lorsqu'une session TCP est répartie sur plusieurs pcaps. Différentes bibliothèques populaires, telles que **libnids**, **libpcap** et **libhttp** offrent des solutions viables à ce problème.

Dans le contexte d'une réponse sur incident, il est notamment intéressant de pouvoir interpréter les commandes et actions effectuées par l'attaquant directement à partir des fichiers pcaps. Le framework Open Source ChopShop [6], développé par le MITRE est par exemple capable de détecter et de décoder le trafic de plusieurs RATs connus.

3 Idées et exemples - Et dans la vraie vie, ça se passe comment ?

3.1 Analyser des comportements

En plus d'une excellente maîtrise de ses outils et techniques d'analyse de flux réseau ou de logs, l'analyste vétérinaire se basera principalement sur son expérience de plusieurs réponses sur incident pour détecter et identifier les pistes, indices ou encore comportements malveillants dans une très grosse quantité d'informations. La connaissance de l'architecture, des applications et des pratiques de l'entreprise compromise (les bonnes, et surtout les mauvaises) seront des atouts essentiels pour détecter des comportements suspects. Une vérification des informations découvertes avec les équipes métiers sera régulièrement nécessaire lors

de l'analyse de l'incident. En effet, l'envoi de 30Go de données vers des systèmes hébergés dans un pays exotique pendant le week-end peut être tout à fait normal ; par exemple un laboratoire qui upload ses résultats à une équipe partenaire.

Ainsi, la recherche de comportements suspects dans les logs et les flux réseau, par un analyste d'expérience, sera séparée en deux parties :

- les marqueurs connus. Passer l'ensemble des marqueurs connus (domaines, IP, URL, user agent, motifs, etc.) disponibles sur des bases publiques ou privées (IOC, marqueurs, liste noire, etc.) ;
- l'inconnu. En parallèle ou après l'étude des marqueurs connus, une analyse plus poussée peut être lancée pour détecter et identifier des pistes, indices ou comportements suspects.

Pour couvrir un maximum de pistes, l'analyste perspicace a souvent recourt à son expérience, à sa veille sur les différentes méthodes d'attaques et à son flair. Cette partie de l'analyse est trop souvent laissée comme un exercice de style, ou les meilleurs pourront exceller et les pires, faire un hors sujet. Il arrive toutefois que l'inverse se produise selon les cas et les connaissances. Sans établir une méthodologie générique d'analyse des pistes et indices qui pourraient contraindre l'analyse et les résultats, cette section propose des idées et axes de recherche non exhaustifs, mais utiles à creuser pour découvrir des comportements suspects [7] [8] :

- *maxima* et *minima*. La recherche de variations dans les maximums et minimums de certaines valeurs ou métadonnées permet de mettre en évidence des comportements illégitimes. Ainsi, l'analyse du top 10 des domaines les plus accédés le week-end, le top 10 des domaines réalisant le plus de volumétrie montante ou les 10 user agent les moins utilisés permettra de lever une alerte si les résultats changent dans le temps, et donneront à l'analyste les premières pistes de recherche ;
- contenu non chiffré sur les ports HTTPS. Bien que cette métrique puisse générer des faux positifs, il n'est pas rare de trouver des malwares communiquant en clair sur les ports véhiculant de manière standard des flux chiffrés comme le port 443. Plus rare encore, l'utilisation d'une connexion SSL avec le *NULL cipher* ;
- *common name* incorrect. Plusieurs architectures de serveurs C&C ou malwares utilisent des certificats possédant des CN non valides. Par exemple, lorsque le CN ne correspond pas aux domaines demandés ou acceptant l'ensemble des domaines possibles (CN=*) ;
- échange de fichiers d'archives sur les ports HTTPS. La détection d'en-têtes de fichiers compressés (RAR, ZIP, 7z) peut être très utile pour découvrir, dans certains cas, une exfiltration de données. Encore une fois, attention aux nombreux faux positifs qui peuvent être générés. Un traitement sur les



fichiers pcap permettra d'extraire a posteriori l'archive ainsi que la liste des fichiers contenus, si l'en-tête n'est pas chiffré ;

- navigation automatisée. La détection d'une navigation Web scriptée n'est pas une tâche facile, mais il existe plusieurs idées pour découvrir rapidement des connexions suspectes. Par exemple, l'absence de téléchargement des scripts JavaScript, CSS ou encore du fichier **favicon.ico** permet de déceler des connexions automatiques possiblement issues de malwares ou de scripts malveillants ;
- informations contenues dans les champs non standards de protocoles réseau. Certains malwares utilisent le champ *user agent* comme canal de contrôle. Pour cela, le calcul d'une simple distribution de cette valeur selon les domaines contactés permettra de découvrir ce type de méthode de communication de ces codes malveillants ;
- géolocalisation des IP sur les accès VPN. Vérifier la géolocalisation des IP sources pour chaque compte VPN permettra de détecter rapidement qu'un agent d'une filiale angevine se connecte également tous les soirs depuis Santiago du Chili. Cette technique est aussi utilisée par des banques pour détecter des fraudes (retraits ou paiements dans plusieurs endroits dans le monde) ;
- méthode HTTP non classique. L'analyse de certaines méthodes HTTP inhabituelles peut fournir des pistes. Par exemple, la méthode PUT est utilisable pour exfiltrer les données contenues dans une archive vers un serveur Web distant ;
- URIs variables. L'objectif est de collecter un ensemble de métriques sur les URIs, selon un domaine donné, pour déterminer si elles sont suffisamment variables et ressemblent à un site web classique (page web, CSS, JavaScript, etc.). En effet, les URIs générés par les malwares comportent, le plus souvent, beaucoup ou peu de variations. La répartition des variations est souvent définie selon une fonction de répartition, où les URIs malveillants sont répartis dans les valeurs extrêmes et les sites web classiques autour de la moyenne. Ainsi, pour étudier les variations, il est intéressant de déterminer pour chaque domaine de deuxième et troisième niveau (*second-level domain* et *third-level domain*) les statistiques suivantes : nombre d'extensions différentes (TLD, gTLD, ccTLD, sTLD), de profondeurs de chemins différents, de noms de fichiers différents et de clefs, valeurs (clef, valeur) différentes pour l'ensemble des path [5] et query [5] de l'URI (par exemple : **foo.php?cle1=valeur1**). L'ensemble de ces statistiques n'est pas exhaustif, mais constitue un moyen rapide de découvrir des domaines ou adresses IP suspectes ;
- fréquence et volume d'envois selon un couple d'e-mail (From et To). La répartition des fréquences et volumes d'envois sont des métriques qui permettent

de trouver des canaux de contrôle utilisant les e-mails ou des boîtes mails compromises utilisées pour envoyer du *spear phishing*. Des outils, comme le périodogramme [9], offrent de bons résultats sur l'étude des intervalles d'envois ou de connexions ;

- temps de réponse d'un e-mail pour une adresse extérieure. Les temps de réponse fixes indiquent des comportements automatiques (robots, e-mails d'absence, e-mails d'erreur, etc.). Les canaux de contrôle, utilisant les e-mails, peuvent être détectés en détectant par exemple des temps de réponse courts et fixes.

3.2 Informations extérieures

Quand cela est possible, les éléments découverts lors du traitement de l'incident peuvent être recherchés et corrélés sur plusieurs bases d'information privées ou publiques. Pour des raisons d'image et de marketing, cette opération de collecte et de corrélation est souvent appelée « Cyber Intelligence ». Les objectifs de la « Cyber Int » (pour les intimes) sont d'obtenir des informations (nouvelles ou historiques) sur le marqueur recherché et confirmer ou infirmer certaines pistes ou hypothèses telles que la famille du malware, le groupe d'attaquants, etc. Cette section se limitera à une rapide introduction des outils et techniques à utiliser dans une réponse sur incident orientée réseau :

- évolutions DNS. Les bases de *passive DNS* [10] conservent les modifications dans le temps des enregistrements DNS liant nom de domaine et adresses IP. L'analyste pourra découvrir grâce à ces bases, d'anciens ou nouveaux domaines et adresses IP liés à un C&C. Ces nouveaux éléments seront ensuite ajoutés dans les sondes ou dans l'analyse des logs pour augmenter l'étendue de détection de l'attaque et trouver de nouvelles machines compromises ;
- enregistrements des domaines. Les bases *Whois* stockent des informations concernant l'enregistrement des noms de domaines. En particulier, il existe des bases enregistrant l'historique des données Whois qui renseigneront l'analyste sur l'évolution de l'enregistrement du domaine analysé. Ainsi, dans certains cas extraire l'adresse e-mail ou les coordonnées de la « personne » qui a enregistré le C&C, pour les recouper avec des enregistrements d'autres domaines, pouvant faire partie de la même attaque ou du même groupe d'attaquant [11] sera utile ;
- géolocalisation des IP. Plusieurs sites ou services proposent, moyennant finances, de fournir ce type d'information. Toutefois, il est conseillé de télécharger des bases de géolocalisation comme Maxmind ou Hostip.info [12] et de réaliser le traitement localement pour des raisons de performance et réduire la divulgation d'informations à un tiers.



- type de domaine. Déterminer comment les domaines contactés par les malwares sont générés peut renseigner sur le type et le niveau de l'attaquant : Cherche-t-il à éviter la détection ou fait-il le strict minimum ? On peut citer différentes techniques comme le DDNS (*Dynamic DNS*), le Fast Flux ou les DGA (Domain Generation Algorithm). Ainsi, il sera utile de garder une liste à jour [13] de l'ensemble des DDNS et ne pas hésiter à étudier, pour chaque domaine de second niveau, l'ensemble des domaines de niveaux suivants. Le calcul de l'entropie de Shannon [14] sera une bonne métrique pour découvrir les domaines générés par des DGA ;
- domaines connexes. Étudier les domaines résolus par un système compromis avant et après une requête vers un C&C, par exemple sur une durée de 60 secondes, peut donner à l'analyste d'autres éléments sur le fonctionnement de l'attaque. Ces domaines nommés Co-occurrences et *Related Domains* [15] ne sont pas forcément malveillants, mais ils peuvent aider à comprendre le fonctionnement des connexions du malware vers le C&C. En effet, il ne sera pas étonnant de détecter dans ces domaines connexes l'enchaînement des différentes étapes de la compromission ou découvrir des requêtes a priori légitimes, mais effectuées par le malware pour tester la connectivité à Internet ;
- configuration du malware. Sans aborder l'ensemble des solutions d'analyses statiques et dynamiques de malware, on notera l'existence de scripts Python disponibles sur le github RATDecoders [16] dont le but est de décoder et d'extraire automatiquement les configurations de plusieurs RAT. Une plate-forme gratuite en ligne est également ouverte [17] ;
- bases de recherches privées ou publiques. Pour finir, on notera que de multiples bases publiques de malwares sont consultables sur Internet : Virus Total Intelligence, malwr.com, Totalhash, Clean-mx, Virusshare, ainsi que l'utilisation d'opérateurs dans les moteurs de recherche classiques [18] permettent de collecter de nouveaux éléments.

Note

La divulgation des éléments d'analyse à un tiers doit être étudiée et contrôlée et en respect avec la classification et le niveau de diffusion de l'information (TLP : RED, AMBER, GREEN et WHITE [19]). Ainsi, il sera toujours préférable de réaliser l'ensemble des traitements sur des outils et bases locaux.

Conclusion

Finalement, l'efficacité d'une analyse de logs et de flux réseau réside dans la capacité de l'analyste à canaliser son expérience, sa démarche et sa sagacité

pour détecter, vérifier et creuser, dans l'ensemble des informations, des indices ou pistes de comportements suspects. Pour cela, cette spontanéité dans l'analyse doit être assurée par l'utilisation d'outils réactifs, simples et flexibles, une préparation des méthodes et outils, avant d'arriver sur le terrain de l'incident. ■

■ RÉFÉRENCES & LIENS

- [1] Conservation pendant 1 an - Article R10-13 (décret n°2006-358 du 24 mars 2006)
- [2] <http://www.cert.ssi.gouv.fr/site/CERTFR-2014-ACT-024> et <http://www.cert.ssi.gouv.fr/site/CERTFR-2014-ACT-025>
- [3] <http://www.ssi.gouv.fr/fr/guides-et-bonnes-pratiques/recommandations-et-guides/securite-du-poste-de-travail-et-des-serveurs/recommandations-de-securite-pour-la-mise-en-oeuvre-d-un-systeme-de.html>
- [4] <http://christophe.vandepas.com/2014/06/setting-up-single-node-elk-in-20-minutes.html>
- [5] RFC 3986 - <http://tools.ietf.org/html/rfc3986>
- [6] <https://github.com/MITRECND/chopshop>
- [7] Bien que cette méthode semble plus souple, les bonnes pratiques et méthodologies forensic doivent être, sans exception, appliquées : traçabilité des actions, main courante, templates d'analyse et de rapport, structuration et consolidation des données dans une solution de partage et d'échange, comme un Wiki.
- [8] En raison des détails de certaines idées ou leurs complexités, l'ensemble des idées ne pourra pas être traité en détail. N'hésitez pas à nous contacter si vous avez des questions ou des meilleures idées.
- [9] https://fr.wikipedia.org/wiki/Estimation_spectrale#Le_p_C3.A9riodogramme
- [10] BFK, CERTEE, DNSParse, DNSDB, VirusTotal
- [11] Ce service (Reverse Whois) est très souvent payant et les informations y sont parfois partielles (les informations Whois peuvent être anonymisées et non renseignées selon les cas).
- [12] https://www.maxmind.com/en/geolocation_landing, <http://www.hostip.info/dl/index.html>
- [13] <http://dnslookup.me/dynamic-dns/>, <https://freedns.afraid.org/domain/registry/>
- [14] https://en.wikipedia.org/wiki/Information_theory#Quantities_of_information
- [15] <http://labs.opendns.com/2013/07/24/co-occurrences/>
- [16] <https://github.com/kevthehermit/RATDecoders>
- [17] <http://malwareconfig.com/>
- [18] <https://sites.google.com/site/gwebsearcheducation/advanced-operators>
- [19] <https://www.us-cert.gov/tlp>

CONTRÔLER LA SÉCURITÉ DES OBJETS DE L'ACTIVE DIRECTORY AVEC BTA

Joffrey Czarny aka _Sn0rkY

Security Researcher at Airbus Group Innovations

joffrey.czarny@airbus.com



mots-clés : SÉCURITÉ / ACTIVE DIRECTORY / DOMAINE & FORÊT WINDOWS / BTA

L'une des premières tâches de l'attaquant, lorsqu'il dispose d'un accès à un réseau d'entreprise, est la compromission de l'Active Directory. Pourquoi ne pas utiliser BTA pour rechercher des backdoors au sein de l'AD ? Les « mauvaises pratiques d'admins » découvertes donneront encore plus de sueurs froides aux équipes de réponse à incident :)

1 Introduction

Lorsqu'il est question de la sécurité du système d'information, les contrôleurs de domaine Active Directory sont, ou devraient être, au centre des préoccupations. En temps normal, pour s'assurer du respect des bonnes pratiques et lors d'une compromission avérée, pour explorer la possibilité d'un nettoyage du système d'information dans l'attente d'une reconstruction. En effet, l'attaquant aura peut-être laissé quelques portes dérobées au sein de l'Active Directory pour en garder le contrôle. Il est donc important de les identifier au même titre que les RAT pour être serein dans le nettoyage.

Nous présentons ici quelques points de contrôle sur l'Active Directory, sans être exhaustifs, pouvant être réalisés avec l'outil BTA [1]. L'analyse des résultats issus de ces points de contrôle pourra permettre d'identifier d'éventuelles modifications malveillantes dans un domaine.

Le conteneur Active Directory situé au niveau supérieur est appelé une « Forêt ». Une Forêt se compose d'un ou de plusieurs domaines ayant en commun un schéma et un catalogue global. Lors de la création du premier contrôleur de domaine d'une organisation, celui-ci est le domaine racine de la Forêt. Certaines organisations peuvent disposer de plusieurs Forêts.

Une forêt est une limite de sécurité et d'administration pour tous les objets qu'elle contient. Un domaine est une limite d'administration destinée à faciliter la gestion d'objets tels qu'utilisateurs, groupes et ordinateurs. De plus, chaque domaine applique ses propres stratégies de sécurité et relations d'approbation avec les autres domaines. Il faut bien comprendre que le périmètre de sécurité dans l'Active Directory se trouve au niveau de la Forêt et non au niveau d'un domaine.

BTA fonctionne par domaine, mais lors d'une analyse, il est important de faire l'analyse au sein de la Forêt.

2 Rappel sur le contrôle d'accès sous Windows (ACLs, DACL, SACL et ACE)

Les protections permettant de définir les autorisations d'accès aux objets dans l'Active Directory reposent sur trois bases :

- les descripteurs de sécurité ;
- l'héritage d'objet ;
- l'authentification de l'utilisateur.

Les descripteurs de sécurité contiennent deux listes de contrôles d'accès (ACL) : la liste de contrôle d'accès discrétionnaire DACL et la liste de contrôle d'accès système SACL.

Une DACL (*Discretionary Access Control List*) identifie les utilisateurs auxquels les autorisations d'accès sont données ou refusées sur l'objet la référençant. Celle-ci contient les entités de contrôle d'accès (ACE, *Access Control Entries*) qui déterminent l'accès d'utilisateur à l'objet considéré. Ainsi la DACL du compte administrateur (objet de type utilisateur dans l'Active Directory) listera l'ensemble des utilisateurs et groupes qui disposent de droits sur le compte administrateur. Si une DACL n'identifie pas explicitement un utilisateur, ou un groupe dont l'utilisateur est membre, l'accès à l'objet lui sera refusé. Par défaut, c'est le propriétaire d'un objet ou son créateur qui contrôle la DACL.

Une SACL (*System Access Control List*) contient les ACE qui déterminent s'il faut journaliser les tentatives d'accès à un objet, réussies ou manquées, effectuées selon une autorisation particulière (contrôle total ou lecture, par exemple).

Les objets Active Directory peuvent hériter des ACE du descripteur de sécurité situé dans l'objet conteneur parent. Grâce au processus d'héritage, les informations de contrôle d'accès définies au niveau d'un objet conteneur dans Active Directory s'appliquent aux descripteurs de sécurité de tout objet subordonné, y compris à d'autres conteneurs et aux objets de ceux-ci. De cette façon, il n'est plus nécessaire de paramétrer des autorisations à chaque création d'objet enfant.

Active Directory se charge également d'authentifier les utilisateurs et les ordinateurs cherchant à accéder à des objets du réseau, cette partie ne sera pas traitée dans l'article [4].

3 Présentation de BTA

L'outil BTA a été développé afin de pouvoir extraire et analyser des informations d'un AD. Il est développé en Python, utilise le moteur MongoDB pour la base de données et fonctionne sous Linux.

Il est conçu autour du concept de *miners*, qui sont des petits bouts de code ayant pour rôle de vérifier un point de contrôle ou une famille de points de contrôle.

BTA transfère chaque base Active Directory dans une base de données.

Les *miners* analysent les informations contenues dans cette base de données et génèrent un rapport sous forme d'une structure de données hiérarchique sommaire, pouvant contenir sections, listes et tables. Ces rapports peuvent être combinés en un seul document. Enfin, ce document peut être traduit en un format de sortie tel que *RestructuredText*, *Excel* ou *CSV*.

BTA propose également un outil permettant de comparer deux bases de données contenant le même Active Directory à deux instants différents.

Le framework est composé de quatre outils :

- **btainport** importe la base **NTDS.dit** dans une base MongoDB et initialise un *postprocessing* ;
- **btamanage** effectue l'intendance sur les bases NTDS importées (lister, effacer...) ;
- **btaminer** appelle les *miners* permettant de requêter les données et propriétés des objets ;
- **btadiff** effectue un différentiel entre deux instances de l'AD intégrées en base.

4 Méthodologie d'audit

4.1 Extraction

La procédure d'extraction du fichier **NTDS.dit** doit être rigoureusement suivie. L'expérience a montré que lorsqu'elle ne l'était pas ou lorsque des raccourcis étaient pris, le fichier **NTDS.dit** résultant était souvent corrompu.

Pour cela, dans un environnement Windows 2008 ou supérieur, nous recommandons l'utilisation de l'outil **ntdsutil** avec la méthode IFM (*Install From Media*). Cette méthode permet d'automatiser la duplication du fichier **NTDS.dit** sans se préoccuper du Volume Shadow Copy Service ainsi que de la synchronisation de la base ESE avec **esentutil.exe**.

Voilà les actions à mener :

```
C:\>ntdsutil
activate instance ntds
ifm
create full c:\NTDS_saved
quit
quit
```

Voici un exemple concret sur un AD 2008 de test en français. Les commandes à taper sont en gras.

```
C:\> ntdsutil
ntdsutil: activate instance ntds
Instance active définie à " NTDS ".
ntdsutil: ifm
ifm : create full c:\NTDS_saved
Création d'une capture instantanée...
Le jeu de captures instantanées {ed7f5b24-82d9-4d30-9ebf-0cf86045d7e9} a été
généré.
Capture instantanée {ca9c5a4c-3450-4bdf-be10-62667f456f38} montée en tant
que C:\$SNAP_201404171405_VOLUMEC$\
La capture instantanée {ca9c5a4c-3450-4bdf-be10-62667f456f38} est déjà
montée.
Initialisation du mode DEFRAGMENTATION...
Base de données source :
C:\$SNAP_201404171405_VOLUMEC$\Windows\NTDS\ntds.dit
Base de données cible :
c:\NTDS_saved\Active Directory\ntds.dit

Defragmentation Status (% complete)
```

```

0 10 20 30 40 50 60 70 80 90 100
|---|---|---|---|---|---|---|---|---|
.....

```

```

Copie de fichiers de Registre...
Copie : c:\NTDS_saved\registry\SYSTEM
Copie : c:\NTDS_saved\registry\SECURITY
Capture instantanée {ca9c5a4c-3450-4bdf-be10-62667f456f38} démontée.
Support IFM créé dans c:\NTDS_saved
ifm : quit
ntdsutil: quit
C:\>

```

Dans un environnement Windows 2003, VSS sera utilisé via l'utilitaire **vssadmin.exe**. Cependant, une fois la copie de l'ensemble du répertoire NTDS effectuée, il est important de fermer et de mettre à jour la base ESE en utilisant la commande **esentutl.exe** dans le répertoire où se trouve la copie du fichier **NTDS.dit** (dans notre exemple **c:\NTDS_saved**).

Voici les commandes à taper sur un serveur Windows 2003 :

```

C:\NTDS_saved> vssadmin create shadow /for=c:
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001 Microsoft Corp.

Successfully created shadow copy for 'c:\'
Shadow Copy ID: {43a58150-a471-49b4-8bc1-1a7d967d32d0}
Shadow Copy Volume Name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1

C:\NTDS_saved> copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\
NTDS} .
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\edb.chk
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\edb.log
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\edbres00001.jrs
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\edbres00002.jrs
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\ntds.dit
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\temp.edb
6 fichier(s) copié(s).

C:\NTDS_saved> esentutl.exe /r edb /d /i /8

Microsoft(R) Windows(R) Database Utilities
Version 5.2
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating RECOVERY mode...
Logfile base name: edb
Log files: <current directory>
System files: <current directory>
Database Directory: <current directory>

Performing soft recovery...

Operation completed successfully in 0.468 seconds.

C:\NTDS_saved> vssadmin delete shadows /for=c:
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001 Microsoft Corp.

Do you really want to delete 1 shadow copies (Y/N): [N]? Y

Successfully deleted 1 shadow copies.

```

Le fichier **C:\NTDS_saved\NTDS.dit** peut maintenant être récupéré et plus aucune action n'est nécessaire sur l'AD.

4.2 Import des données

Une fois le fichier **NTDS.dit** récupéré, la commande **btainport** va se charger d'importer les données dans une base MongoDB.

Suite à l'import des données, de nouvelles collections sont créées dans la base MongoDB afin de pré-calculer certaines valeurs telles que la liste des catégories de classe de schémas *schemaIDGUID*, d'objets *objectGUID*, de droits *rightsGuid*, qui pourront être utilisées par les miners afin de simplifier et d'accélérer certaines requêtes.

```

$ btainport -C ::test2 mon.ntds.dit
INFO : Going to import ::test2      <- mon.ntds.dit
INFO : Opening [mon.ntds.dit]
INFO : Opening done.
INFO : ### Starting importation of sd_table ###
Importing [test2.sd_table]: 156 / 156
-- avg=277.75 rec/s inst=217.11 rec/s
-- ETA=0:00:00 elapsed=0:00:00
INFO : done.
INFO : Creating metatable
INFO : Importation of sd_table is done.
INFO : ### Starting importation of link_table ###
Importing [test2.link_table]: 1212 / 1212
-- avg=6327.75 rec/s inst=4767.61 rec/s
-- ETA=0:00:00 elapsed=0:00:00
INFO : done.
INFO : Creating metatable
INFO : Importation of link_table is done.
INFO : ### Starting importation of datatable ###
INFO : Resolving column names
Scanning for column names: 7500 / 7631
-- avg=3500.28 recs/s inst=3343.46 recs/s
-- ETA=0:00:00 elapsed=0:00:02
INFO : All columns found! Ending scan early!
INFO : Resolved 3200 / 3200 columns.
Importing [test2.datatable]: 7631 / 7631
-- avg=59.37 rec/s inst=59.90 rec/s
-- ETA=0:00:00 elapsed=0:02:08
INFO : done.
INFO : Creating metatable
INFO : Importation of datatable is done.
INFO : Post-processing: category
INFO : Post-processing: domains
INFO : Post-processing: dnames
INFO : Post-processing: usersid
INFO : Post-processing: rightsGuids

```

4.3 Analyse et investigation au sein de l'AD via l'exécution des miners

Lors d'une analyse de sécurité de l'Active Directory récurrente ou dans le cas d'une compromission avérée, il est important d'auditer les permissions positionnées sur des objets spécifiques, tels que *AdminSDHolder*,

les permissions accordées à des utilisateurs, telles que *User-force-Change-Password* ou bien les attributs d'un objet, tel que *SIDHistory*. Ces points de contrôle permettent d'avoir une première vision sur la sécurité des objets de l'AD et du niveau de confiance que l'on peut avoir dans les contrôles d'accès opérés par l'AD.

Ci-dessous, voici les quelques points de contrôle qui vous seront présentés et dont la maîtrise a un grand intérêt en matière de sécurité :

- les groupes sensibles *Built-in* et métiers ;
- l'intégrité d'*AdminSDHolder* et de la fonctionnalité *SDProp* ;
- les droits positionnés sur le schéma Active Directory ;
- l'attribution de droits étendus ;
- les contrôles d'accès positionnés sur les boîtes e-mail.

Il ne faut cependant pas oublier que l'Active Directory peut disposer d'un fort historique de configuration. Certaines interrogations ne pourront être résolues sans un échange avec les administrateurs de l'Active Directory. De nombreux autres points de contrôles devraient être effectués, mais dépendent de la connaissance du fonctionnement intime de l'Active Directory et des extensions de schéma liés à l'intégration d'*Exchange* ou *SharePoint*. Typiquement, peu de *miners* ont été développés pour analyser la sécurité des objets *Exchange*, idem pour *SharePoint*.

4.4 Vérification sur les membres des groupes sensibles/Built-in

Commençons par l'analyse des groupes sensibles : Administrateurs de l'entreprise, Administrateurs du domaine et groupes sensibles d'un point de vue métier, fournis par l'administrateur.

L'ensemble des membres des groupes sensibles doit être connu et maîtrisé par les administrateurs de l'Active Directory. Il est également important de valider que les ACE positionnées sur chacun de ces membres soient conformes afin de déterminer si l'un des comptes ne peut pas être compromis par une lacune ou un droit trop permissif.

Pour cela, nous utiliserons le *miner ListGroup*, qui effectue deux tâches :

- lister l'ensemble des membres du groupe, en nous indiquant :
 - ↳ si le compte n'est plus membre,
 - ↳ les *flags* *UserAccountControl*,
 - ↳ son appartenance à un sous-groupe membre.

```
~$ btaminer -C ::snktest -t ReST ListGroup --match "Admins du domaine"

Analysis by miner: [ListGroup]

=====

List of groups matching [Admins du domaine]

Group Admins du domaine
-----

sid = S-1-5-21-1154779122-356131834-6750385761-512
guid = 8bfff35e5-87ff-4d9f-b979-122adf32cdd9
dn = .snk.snowcrash.metaverse.Users.Admins du domaine
+-----+-----+-----+-----+
| User | Deletion | Flags | Recursive |
+-----+-----+-----+-----+
| Hiro Protagonist | | normalAccount, smartcardRequired, dontExpirePassword | |
| Juanita Marquez | | normalAccount, smartcardRequired, dontExpirePassword | |
| Administrateur | | normalAccount, accountDisable | |
| Raven | 13-04-01 | normalAccount | |
+-----+-----+-----+-----+

[...]
```

Dans l'exemple ci-dessous, il est possible d'identifier que le compte *Raven* a été membre du groupe Admins du domaine. Ceci peut être la trace d'une élévation de privilèges temporaire lors d'une attaque. Cette information provient de la table **link_table** via l'attribut '**link_delttime**'. Cette valeur n'est pas accessible via LDAP ou ADSI, d'où l'intérêt d'extraire les données du fichier **NTDS.dit**. De plus, on voit que le compte *Raven* ne dispose pas du flag UAC *UserAccountControl* « smartcard_required » comme les autres comptes membres du groupe Admins du domaine.

- lister les ACE sur chacun des membres.

```
User Hiro Protagonist (S-1-5-21-1154779122-356131834-6750385761-1337)

+-----+-----+-----+-----+
| Trustee | Member | ACE Type | Object type |
+-----+-----+-----+-----+
| Admins du domaine | Hiro Protagonist | AccessAllowedObject | (none) |
| Everyone | Hiro Protagonist | AccessAllowedObject | User-Change-Password |
| Raven | Hiro Protagonist | AccessAllowedObject | (none) |
| Self | Hiro Protagonist | AccessAllowedObject | User-Change-Password |
| Self | Hiro Protagonist | AccessAllowedObject | Private-Information |
| Admins du domaine | Hiro Protagonist | AccessAllowed | (none) |
| Administrateurs | Hiro Protagonist | AccessAllowed | (none) |
| System | Hiro Protagonist | AccessAllowed | (none) |
| [...] |
| Everyone | Hiro Protagonist | SystemAudit | (none) |
| Everyone | Hiro Protagonist | SystemAuditObject | GP-Link |
| Everyone | Hiro Protagonist | SystemAuditObject | GP-Options |
+-----+-----+-----+-----+
```

Dans l'extrait ci-dessus, issu de la commande précédente, le compte *Raven* dispose de l'ensemble des privilèges sur le compte *Hiro Protagonist*. L'*objectType* sert à spécifier selon les cas un droit étendu, une écriture validée, une propriété, une classe d'objet, etc. Quand il est vide comme ici, cela signifie que tous les éléments sont affectés. Mais ces éléments dépendent des droits spécifiés dans un champ de l'ACE

appelé **AccessMask**. Ici, pour des raisons de visibilité le contenu de l'**AccessMask** n'a pas été affiché, il est cependant impératif de contrôler cette valeur pour connaître les droits spécifiés.

4.5 Vérification sur des comptes compromis

Dans le cas d'une compromission avérée, une liste de comptes ayant été compromis et utilisés par l'attaquant pourrait être fournie par les équipes de réponses à incident. Dans ce cas, il est intéressant d'étudier les modifications relatives sur ces comptes et leur adhérence avec les autres objets de l'AD. Pour cela, le *miner* :

- *ListACE* pourra être utilisé avec l'option **trustee** afin de lister les droits dont dispose le compte ;

```
~$ btaminer -C ::snktest -t ReST ListACE -trustee S-1-5-21-1154669122-758131934-2550385761-1666
```

- **InvestUser** pourra être utilisé afin de lister les valeurs des attributs **lastLogon** et **logonCount** qui donneront une indication sur l'utilisation du compte, celle de l'attribut **pwdLastSet** la date du dernier changement de mot de passe et enfin celle de **badPasswordTime** la date à laquelle une authentification sur l'AD a échoué.

```
~$ btaminer -C ::snktest -t ReST InvestUser --match "Raven"

Analysis by miner: [InvestUser]
=====

sid = S-1-5-21-1154669122-758131934-2550385761-1666
dn = .snk.snowcrash.metaverse.Users.Raven

Last Logon time: 2014-07-24T11:59:48.905Z.
This account's password has been last modified on
2014-07-02T14:16:17.625Z.
Logon Count is set to "5979"
badPasswordTime is set to "2014-06-13T04:40:41.390Z".
```

Ces informations en l'état ne pourront pas forcément fournir une analyse complète des activités de l'attaquant, mais, couplées avec d'autres événements de l'AD, elles permettront potentiellement de chaîner ou confirmer des actions malveillantes. Le changement de mot de passe d'un compte d'administration en pleine nuit ou l'utilisation d'un compte de service dans une période inhabituelle peut indiquer des actions suspectes.

4.6 Vérification de l'objet AdminSDHolder

AdminSDHolder [7] peut être vu comme une sauvegarde de descripteur de sécurité pour les groupes et utilisateurs privilégiés qu'il protège, via le

processus SDPROP, d'une modification erronée. Il est donc essentiel de contrôler les permissions relatives à *AdminSDHolder*, afin d'identifier une éventuelle porte dérobée liée à une permission incorrecte ou volontairement positionnée.

L'objet *AdminSDHolder* a une liste unique de contrôle d'accès (ACL) qui est utilisée pour contrôler les autorisations d'entités de sécurité qui sont membres de groupes privilégiés *Built-in* dans Active Directory (les groupes « protégés »). Chaque heure, un processus en tâche de fond s'exécute sur le contrôleur de domaine. Il compare l'ACL des entités de sécurité (utilisateurs, groupes et comptes d'ordinateur) qui appartiennent à des groupes protégés contre l'ACL de l'objet *AdminSDHolder*. Si la taille ou la chaîne binaire est différente, le descripteur de sécurité de l'objet est remplacé par le descripteur de sécurité de l'objet *AdminSDHolder*.

La valeur de l'attribut **AdminCount** d'un compte utilisateur passe de 0 à 1 lorsqu'on lui accorde des privilèges d'administration (Opérateur d'impression, Admins de domaine, ...) au sein de l'Active Directory. Cette valeur ne change pas, et reste à 1, lorsque l'on révoque ces privilèges. Ainsi, en listant l'ensemble des comptes disposant de l'**AdminCount** à 1, il est possible de savoir si un compte a eu des privilèges d'administration.

Deux points doivent donc être analysés : la liste des contrôles d'accès répliqué par *AdminSDHolder* ainsi que les comptes protégés et qui disposent de l'attribut **AdminCount** à 1.

Ci-dessous, via le *miner* **SDProp** et l'option « **CheckACE** » deux anomalies peuvent être identifiées.

```
~$ btaminer -C ::snktest -t ReST SDProp --checkACE
```

```
Analysis by miner: [SDProp]
=====
```

cn	type	SID
Accès compatible pré-Windows 2000	AccessAllowed	ALL
Administrateurs	AccessAllowed	ALL
Administrateurs de l'entreprise	AccessAllowed	ALL
Admins du domaine	AccessAllowedObject	User
Admins du domaine	AccessAllowed	ALL
Everyone	AccessAllowedObject	User-Change-Password
Everyone	SystemAudit	ALL
Everyone	SystemAuditObject	Organizational-Unit
Everyone	SystemAuditObject	Organizational-Unit
Exchange Servers	AccessAllowedObject	User
Exchange Servers	AccessAllowedObject	DS-Replication-Synchronize
Exchange Servers	AccessAllowedObject	Exchange-Personal-Information
Exchange Servers	AccessAllowedObject	Canonical-Name
[...]		
Exchange Servers	AccessAllowedObject	ms-Exch-Safe-Senders-Hash
L. Bob Rife	AccessAllowedObject	ms-Exch-Mailbox-Security-Descriptor
Exchange Trusted Subsystem	AccessAllowedObject	Proxy-Addresses
Exchange Trusted Subsystem	AccessAllowedObject	Show-In-Address-Book
Exchange Trusted Subsystem	AccessAllowedObject	Exchange-Personal-Information

Exchange Trusted Subsystem	AccessAllowedObject	Admin-Display-Name
[...]		
Exchange Trusted Subsystem	AccessAllowedObject	Legacy-Exchange-DN
Exchange Trusted Subsystem	AccessAllowedObject	ms-Exch-Dynamic-Distribution-List
Exchange Trusted Subsystem	AccessAllowed	ALL
Groupe d'accès d'autorisation Windows	AccessAllowedObject	Token-Groups-Global-And-Universal
Raven	AccessAllowedObject	User
Network Service	AccessAllowedObject	Exchange-Personal-Information
Organization Management	AccessAllowedObject	Proxy-Addresses
Organization Management	AccessAllowedObject	Show-In-Address-Book
Organization Management	AccessAllowedObject	Exchange-Personal-Information
[...]		
Organization Management	AccessAllowedObject	ms-Exch-Dynamic-Distribution-List
Organization Management	AccessAllowed	ALL
S-1-5-11	AccessAllowedObject	Exchange-Information
S-1-5-11	AccessAllowed	ALL
Self	AccessAllowedObject	User-Change-Password
Self	AccessAllowedObject	Private-Information
System	AccessAllowed	ALL
Éditeurs de certificats	AccessAllowedObject	X509-Cert

Le compte *L. Bob Rife* dispose des droits sur l'attribut **ms-Exch-Mailbox-Security-Descriptor** qui permet de changer les ACL de toutes les boîtes e-mails.

Le compte *Raven* quant à lui, dispose de droits sur l'objet **User**, ce qui n'est pas logique et qui lui permettrait de changer l'un des *attributs* (et pas les ACE) de cet objet, qui sera alors appliqué à chaque nouvel utilisateur créé.

4.7 Vérification sur l'innocuité du schéma

Pour analyser la santé du schéma, plusieurs contrôles sont à effectuer.

Premièrement, il faut vérifier qui est le propriétaire du Schéma « Administrateurs du Schéma : S-1-5-21 SID : *domaine racine*-518) via les valeurs de **nTSecurityDescriptor** (champ **Owner** et droit **WRITE_OWNER**).

```
~$ btaminer -C ::snktest -t ReST Schema --owner
```

Analysis by miner: [Schema]

Name	SID	Number	Schema type
Administrateurs du schéma	S-1-5-21-1154669122-758131934-2550385761-518	1	Schema
Administrateurs du schéma	S-1-5-21-1154669122-758131934-2550385761-518	4133	Attribute-Schema

Ci-dessous, le résultat est cohérent, seul l'administrateur du schéma est le propriétaire du schéma et des objets attributs de schéma.

Il faut également contrôler les dates de création et de modification des objets **ClassSchema** et **AttributeSchema**.

Pour lister par date le nombre d'objets Schéma créés :

```
$ btaminer -C ::snktest -t ReST Schema --timelineAS created
```

Analysis by miner: [Schema]

Date	Affected attribute schema
2009-02-11 18	1314
2011-12-20 00	2117
2011-12-22 14	61
2011-12-23 18	91

```
$ btaminer -C ::snktest -t ReST Schema --timelineCS created
```

Analysis by miner: [Schema]

Date	Affected class schema
2009-02-11 18	234
2011-12-20 00	267
2011-12-22 14	3
2011-12-23 18	46

Pour lister par date le nombre d'objets Schéma modifiés :

```
$ btaminer -C ::snktest -t ReST Schema --timelineCS changed
```

Analysis by miner: [Schema]

Date	Affected class schema
2009-02-11 18	214
2011-12-20 00	276
2011-12-22 14	10
2011-12-23 18	50
2012-04-01 00	1

```
$ btaminer -C ::snktest -t ReST Schema --timelineAS changed
```

Analysis by miner: [Schema]

Date	Affected attribute schema
2009-02-11 18	1269
2011-12-20 00	2153
2011-12-22 14	69
2011-12-23 18	92

Les dates doivent être dans les périodes correspondantes à l'extension du schéma (installation Exchange, SCOM, etc.).

Il n'est pas logique d'avoir un seul attribut de schéma modifié. Un scénario d'attaque pourrait être de modifier l'attribut **defaultSecurityDescriptor** qui est un attribut des objets de classe '**Class-Schema**' contenant le descripteur de sécurité par défaut qui sera affecté aux objets de cette classe à leur création.

4.8 Vérification des SIDHistory

L'attribut **SIDHistory** est utilisé lors de migrations de domaine pour conserver les accès aux ressources d'un domaine Active Directory. Cela simplifie la migration en évitant d'avoir à modifier les ACL sur les objets et ressources.

Cependant, et malgré la fonction de *SID Filtering* dont le but est de protéger contre les approbations externes sortantes un attaquant ayant pris le contrôle d'un domaine pour tenter d'élever ses privilèges en utilisant l'attribut **SIDHistory** pour prendre le contrôle de la Forêt. Par exemple, le SID « S-1-5-21-domain-519 » Administrateurs de l'entreprise n'est pas filtré en intra Forêt [4]. Ci-dessous, un exemple concret de ce que je ne vous souhaite jamais de trouver ; qui impliquerait une action clairement malveillante au sein de l'AD que vous analysez.

```
$ btaminer -C ::snktest -t ReST SIDHistory --list
Analysis by miner: [SIDHistory]
=====
+-----+-----+
| Name | SIDHistory |
+-----+-----+
| Mr. Lee | S-1-5-21-1112191652-3018143026-3146074675-2050 |
| L. Bob Rife | S-1-5-21-5692136444-8563145266-2513671054-519 |
| Mr. NG | S-1-5-21-1112191652-3018143026-3146074675-2055 |
| DaSid | S-1-5-21-1112191652-3018143026-3146074675-2082 |
+-----+-----+
```

Le compte *L. Bob Rife* est admin de la Forêt alors qu'il est simple utilisateur.

4.9 Vérification sur les droits étendus spécifiques

Certains droits étendus doivent être contrôlés, tels que *User-Force-Change-Password* (type : 00299570-246d-11d0-a768-00aa006e0529), qui permet à celui qui le possède de réinitialiser le mot de passe du compte sans connaître le mot de passe d'origine.

Un attaquant pourrait octroyer ce droit à un compte dont le mot de passe n'expire jamais et ainsi pouvoir réinitialiser les mots de passe d'un membre du groupe « Domain admins », par exemple.

Il faut également être attentif au droit *Self-Membership* (type bf9679c0-0de6-11d0-a285-00aa003049e2) qui permet de modifier l'appartenance d'un compte à un groupe pour le bénéficiaire de l'ACE.

Ci-dessous, un exemple du résultat du *miner* ListACE, permettant d'identifier que le compte *Raven* a le droit de réinitialiser le mot de passe du compte Administrateur.

```
$ btaminer -C ::snktest -t ReST ListACE --type 00299570-246d-11d0-a768-00aa006e0529
Analysis by miner [ListACE]
=====
+-----+-----+-----+
| Trustee | Subjects | Object type |
+-----+-----+-----+
| Raven | Administrateur | User-Force-Change-Password |
+-----+-----+-----+
```

4.10 Vérification sur les droits des boîtes e-mail

Les boîtes mails de certaines personnes peuvent contenir des informations sensibles, non chiffrées malheureusement. Afin d'identifier les personnes disposant de droits sur plusieurs boîtes mail avec des accès directs, sans parler de délégation effectuée au travers des attributs : **ms-Exch-Public-Delegates**, **publicDelegates**, **ms-Exch-Public-Delegates-BL**, **publicDelegatesBL**, une requête doit être menée sur les ACE positionnées sur Exchange via l'attribut **msExchMailboxSecurityDescriptor** en éliminant les utilisateurs légitimes tels que le propriétaire de la boîte '**Self**' **S-1-5-10**.

Ci-dessous, un cas concret où l'utilisateur *L. Bob Rife* dispose de l'ensemble des droits sur la boîte mail de « Juanita Marquez », ce qui n'est pas forcément légitime.

```
~$ btaminer -C ::snktest -t ReST MailBoxRights --userid jmarquez
Analysis by miner [MailBoxRights]
=====
Who can access this Mailbox
+-----+-----+
+-----+-----+
+-----+-----+
| Exchange Administrators | {Juanita Marquez [u'+ReadControl']} |
| L. Bob Rife | {Juanita Marquez [u'+WriteOwner', u'+ReadControl', u'+Synchronize', u'+Delete']} |
+-----+-----+
```

Un autre point de contrôle serait par exemple, de regarder les derniers accès des administrateurs sur les boîtes mails via les attributs : **msExchMailboxAuditLastAdminAccess**, **msExchMailboxAuditLastDelegateAccess**. Ce *miner* n'a pas encore été développé malheureusement.

4.11 Analyse différentielle entre deux instances de fichier NTDS

Une analyse différentielle entre deux instances de l'Active Directory avec **btadiff** pourra également être instructive lors d'une analyse forensic. Vu le nombre

d'objets dans l'Active Directory, cette analyse est très fastidieuse, cependant un *quick win* peut être de filtrer certains champs comme le **ref_count** sur la table **sd_table** :

```
~$ btadiff --CA ::ntds_janvier_2014 --CB ::ntds_juillet_2014 --only sd_table -X
sd_refcount

=====
Starting diffing sd_table
-----
B, 40: []
B, 208: []
B, 257: []
B, 1922: []
B, 2107: []
B, 2158: []
B, 2251: []
B, 2313: []
B, 2318: []
B, 2323: []
B, 2328: []
[.....]
A, 49472: []
A, 49473: []
A, 49474: []
A, 49475: []
A, 49476: []
A, 49477: []
-----
Table [sd_table]: 7873 records checked, 2614 disappeared, 2900 appeared, 0 changed
```

Cette requête a permis de constater qu'aucune ACE n'a été modifiée, car la liste entre crochets est vide. Seules de nouvelles ACEs ont été créées.

5

Les limites de l'exercice

Comme présenté tout au long de cet article, seules les informations provenant du fichier **NTDS.dit** sont analysées, ce qui n'est pas suffisant pour l'investigation lors d'une compromission.

En effet, le périmètre doit être étendu et les potentiels comptes locaux sur les serveurs et stations de travail doivent également être revus. Une problématique souvent rencontrée est la gestion des comptes d'administration locaux disposant du même mot de passe sur l'ensemble du parc. De même, l'utilisation de comptes à forts privilèges, au travers d'ouvertures de sessions journalières sur des postes ayant accès à Internet. Ces mauvaises pratiques ne pourront pas être identifiées avec BTA. Il faut donc initier un audit général en sus, afin de contrôler l'ensemble du périmètre.

Je recommande la lecture du papier [5] [6] « Chemin de contrôle de l'Active Directory » qui donne beaucoup d'informations quant à la visibilité des chemins de compromission possibles.

Je rappelle qu'il est également important d'échanger régulièrement avec les administrateurs de l'Active Directory qui pourront lever une hypothèse d'intrusion en expliquant qu'une mauvaise pratique d'administration en est l'origine.

Conclusion

BTA est un outil d'audit destiné à donner en temps contraint un résultat reproductible. Il est également possible de l'utiliser dans le cas d'une analyse forensic afin d'identifier des problèmes liés à une compromission. Cependant, l'Active Directory étant un vaste sujet, il ne faut pas se contenter de passer quelques outils d'analyses, mais réellement creuser l'utilisation de certains attributs qui peuvent amener à des backdoors dans l'AD.

Il est donc certain que nous venons seulement d'aborder qu'une petite partie des contrôles à effectuer. BTA dans sa version actuelle ne dispose pas de tous les *miners* et que beaucoup de travail reste à faire. ■

■ REMERCIEMENTS

Je tiens à remercier une fois de plus l'ensemble de mes collègues pour tout un tas de raison !

Santé Bonheur :)

■ RÉFÉRENCES

- [1] <https://bitbucket.org/iwseclabs/bta>
- [2] https://www.sstic.org/media/SSTIC2014/SSTIC-actes/BTA_Analyse_de_la_securite_Active_Directory/SSTIC2014-Article-BTA_Analyse_de_la_securite_Active_Directory-czarny_biondi.pdf
- [3] https://www.sstic.org/media/SSTIC2014/SSTIC-actes/BTA_Analyse_de_la_securite_Active_Directory/SSTIC2014-Slides-BTA_Analyse_de_la_securite_Active_Directory-czarny_biondi_1.pdf
- [4] https://www.sstic.org/media/SSTIC2014/SSTIC-actes/secrets_dauthentification_pisode_ii_kerberos_cont/SSTIC2014-Article-secrets_dauthentification_pisode_ii_kerberos_contre-attaque-bordes_2.pdf
- [5] https://www.sstic.org/media/SSTIC2014/SSTIC-actes/chemins_de_controle_active_directory/SSTIC2014-Article-chemins_de_controle_active_directory-gras_bouillot.pdf
- [6] <https://github.com/ANSSI-FR/AD-control-paths>
- [7] <http://policelli.com/blog/archive/2009/11/06/understanding-adminsdholder-and-protected-groups/>

SYSTÈMES BSD



Le guide
pour installer
et bien
administrer
ses systèmes
BSD

**DISPONIBLE CHEZ VOTRE
MARCHAND DE JOURNAUX ET SUR :**
boutique.ed-diamond.com



**10th Edition of
the Infosec Conference
21-24 October 2014**



