



LA RÉPONSE D'INCIDENT

**Analyser et réagir
face à une cyberattaque**

**ADRIEN VOISIN
GUILLERME DUVILLIÉ**

LA RÉPONSE D'INCIDENT

Collection informatique

BROOKS J.C., GROW C., CRAIG P., SHORT D., *Cybersécurité. Sécurisation des systèmes informatiques*

DESGRAUPES B., *LaTeX. Apprentissage, guide et référence*, 3^e éd.

GRIMES R.A., *Hacking et contre-hacking. La sécurité informatique*

LUBANOVIC B., *Python. Comprendre les bases et maîtriser la programmation*

MONK S., *Programmation Arduino. Développez rapidement vos premiers programmes*

TAILLET R., *Bien débiter en LaTeX*

TAILLET R., *Python pour la physique. Calcul, graphisme, simulation*

THIRY T., *Les pratiques de l'équipe agile. Définissez votre propre méthode*, 2^e éd.

Chez le même éditeur

AGASHE A., DETROJA P., MEHTA N., *Blockchain : bulle ou révolution ?*

BAISSAIT V., *Déjouer les arnaques en ligne*

BHARGAVA A., *Les algorithmes*

CYPEL A., *Au cœur de l'intelligence artificielle. Des algorithmes à l'IA forte*

CYPEL A., *Voyage au bout de l'IA. Ce qu'il faut savoir sur l'intelligence artificielle*

SALMANDJEE Y., *Utiliser l'IA en 90 cas concrets pour se faciliter la vie*

TUCKFIELD B., *Au cœur des algorithmes. Les bases de la programmation avec Python*

INFORMATIQUE

LA RÉPONSE D'INCIDENT

Analyser et réagir face
à une cyberattaque

**ADRIEN VOISIN
GUILLERME DUVILLIÉ**

Pour toute information sur notre fonds et les nouveautés dans votre domaine de spécialisation, consultez notre site web : **www.deboecksuperieur.com**

Ce livre ne peut être reproduit ni utilisé à des fins d'entraînement de systèmes d'intelligence artificielle. La fouille de textes et de données est interdite conformément à l'article 4(3) de la Directive (UE) 2019/790.

Dans le cas où le livre comporte des ressources numériques en ligne, ces dernières sont des compléments offerts à l'achat du livre.

Par respect du droit d'auteurs, ces ressources sont inaccessibles, interdites de reproduction et réservées à un usage strictement personnel. Elles sont soumises aux conditions d'utilisation en vigueur (CGU) accessibles sur notre site et peuvent, pour des raisons commerciales, légales, d'évolution technique ou de contenu, être supprimées ou interrompues à tout moment, sans garantie de maintien ou de compensation. Dans le cas où les QR codes et liens hypertextes permettent d'accéder à des sites internet tiers, la responsabilité de De Boeck Supérieur n'est pas engagée, notamment quant à leur éventuel dysfonctionnement ou à leur indisponibilité d'accès.

© De Boeck Supérieur s.a., 2026
Rue du Bosquet, 7, B-1348 Louvain-la-Neuve

Tous droits réservés pour tous pays.

Il est interdit, sauf accord préalable et écrit de l'éditeur, de reproduire (notamment par photocopie) partiellement ou totalement le présent ouvrage, de le stocker dans une banque de données ou de le communiquer au public, sous quelque forme et de quelque manière que ce soit.

Dépôt légal :
Bibliothèque nationale, Paris : mai 2026
Bibliothèque royale de Belgique, Bruxelles : 2026/13647/090

ISBN : 978-2-8073-6979-5

Table des matières

Table des matières	5
1 Introduction	9
1 À qui s'adresse ce livre	11
2 Les thématiques abordées dans ce livre	12
3 Les sujets qui ne seront pas discutés dans ce livre	12
4 Organisation du livre	13
5 Supports pédagogiques	13
 I Indicateurs de compromission	 15
2 Introduction	17
3 Indicateurs de compromission Windows	19
1 Le registre Windows	19
2 Event Logs	22
3 Processus et services	27
4 Activité réseau et connexions à distance	34
5 Exécution des applications	39
6 Interpréter les caches d'exécution des applications	46
7 Tâches planifiées	46
8 NTFS, journalisation et métadonnées	51
9 Historique de la navigation web	57
 4 Indicateurs de compromission Linux	 61
1 Les informations système	61
2 Recherche des indices de compromission	90
 II Récolte de preuves numériques	 119
5 Introduction	121
1 Capture de la mémoire vive	121
2 Capture d'une image disque	121
3 Machines virtuelles	122
 6 Récolte et capture de preuves sous Windows	 123
1 PowerShell et WinRM	123
2 Récolte à l'aide d'agents avec Velociraptor	138
3 Capture de la mémoire vive	143
4 Capture d'une image disque	145

7	Récolte et capture de preuves sous Linux	147
1	Bash et SSH	147
2	Récolte à l'aide d'agents avec Velociraptor	159
3	Capture de la mémoire vive	161
4	Capture d'une image disque	168
III	Analyse d'un incident	177
8	Simulation d'un cyberincident	179
1	Scénario	179
2	Architecture de test	180
3	Méthodologie	181
4	Techniques et outils utilisés	183
9	Analyse de la machine Windows chiffrée	185
1	Préambule	185
2	Analyse de la mémoire vive	185
3	Accès aux fichiers récents et exécutions potentielles	193
4	Analyse des journaux Windows	195
5	Analyse de l'image disque	199
6	Bilan	201
10	Analyse du serveur d'annuaire	203
1	Préambule	203
2	Analyse de la mémoire vive	203
3	Analyse des journaux Windows	204
4	Analyse de l'image disque	207
5	Bilan	208
11	Analyse du trafic réseau	211
1	Triage	211
2	Extraction des fichiers	214
3	Analyse manuelle	216
4	Bilan	219
12	Analyse du serveur web	221
1	Préambule	221
2	Analyse de la mémoire vive	221
3	Analyse des journaux	233
4	Analyse du disque dur	238
5	Bilan	240
13	Analyse du serveur de base de données	241
1	Préambule	241
2	Analyse de l'hôte	241
3	Bilan	251

14 Synthèse des résultats	253
15 L'analyse de logiciels malveillants	255
1 Classification	255
2 Méthodologie d'analyse	257
3 Mise en place d'un environnement d'analyse	259
4 Le réseau	260
5 Les bacs à sable	261
6 Les techniques d'évasion à l'analyse	261
7 Calcul de l'empreinte numérique et triage	262
8 Analyse statique de base	263
9 Analyse statique avancée	281
10 Émulation d'un logiciel malveillant	311
11 Analyse dynamique de base	313
12 Analyse dynamique avancée	324
13 Synthèse des résultats	334
14 Conclusion	335
16 Conclusion	337
 IV Annexes	 339
A Pour aller plus loin	341
1 Partie 1	341
2 Partie 2	341
3 Partie 3	342
B Récolte et capture de preuves sous Linux	343
1 scrap.sh	343
2 collect_infos.sh	348
 Liste des extraits de code	 351

1 Introduction

En 1986, Clifford Stoll, un astrophysicien reconverti en administrateur système travaillant pour l'université de Berkeley, détecte une anomalie dans la comptabilité de leur parc informatique. En effet, à cette époque, l'accès à des instances de calcul était fortement limité et chaque utilisation d'un ordinateur par un chercheur était facturée à son département. Cliff remarque un décalage de 75 centimes entre le temps d'utilisation des ordinateurs et ce qui a été réellement facturé. Il commence par suspecter un bug dans les logiciels utilisés pour comptabiliser le temps d'utilisation mais arrive à la conclusion que ceux-ci fonctionnent correctement. Il finit par identifier des activités étranges sur l'une de ses machines. Un utilisateur, connu pour être à l'étranger et ne plus travailler dans ces laboratoires, se connecte à intervalles réguliers. De plus, certaines actions semblent être effectuées avec des accès privilégiés. Cliff identifie le canal par lequel cet utilisateur suspect se connecte et décide d'écouter le trafic transitant par celui-ci. Il découvre, stupéfait, qu'une personne se connectant depuis l'extérieur du campus, a réussi à récupérer des mots de passe d'autres utilisateurs et a également trouvé une technique pour élever ses privilèges sur les machines¹. C'est le début d'une investigation de plusieurs mois durant laquelle Cliff va méticuleusement noter chaque action réalisée par l'attaquant mais également documenter ses différents échanges avec les services de télécommunications, les agents du FBI, de la CIA et de la NSA. Cette enquête surprenante que raconte Cliff Stoll dans son célèbre livre : *The Cuckoo's Egg*², illustre en réalité ce que l'on pourrait appeler aujourd'hui, une *réponse d'incident*. En effet, en identifiant une activité malveillante dans son infrastructure et en tentant de comprendre la portée et l'auteur de cette attaque tout en tenant un journal de bord, on peut dire qu'il s'inscrivait directement dans une méthodologie d'investigation numérique.

À cette époque, Internet n'était encore qu'à ses prémices et les machines interconnectées dont fait mention Cliff Stoll étaient essentiellement utilisées par les universités et l'armée. Ces infrastructures n'ont pas été conçues avec pour objectif de les rendre très sécurisées. Il était relativement facile pour un utilisateur de pouvoir tenter de se connecter sur n'importe quelle machine appartenant à un réseau. En général, un simple couple identifiant/mot de passe protégeait l'accès et il était souvent facile de deviner ceux-ci. Aujourd'hui, on peut dire que la sécurité informatique s'est quelque peu renforcée et que les accès ne sont plus aussi facilement compromis qu'à l'époque. Par contre, la surface d'attaque s'est nettement étendue. De quelques machines utilisées dans des campus et bases militaires, nous retrouvons aujourd'hui des ordinateurs connectés partout dans le monde, que ce soit dans la sphère privée ou dans le monde professionnel. Les infrastructures critiques (santé, défense, énergie, administrations) sont toutes dépendantes de l'informatique et à une connectivité constante. La plupart des personnes utilisent des téléphones intelligents et les objets connectés se multiplient. En réalité, l'économie et les modes de fonctionnement de notre société actuelle reposent en grande partie sur l'informatique et sur Internet en particulier. Il n'est donc pas surprenant qu'une nouvelle forme de criminalité se soit développée en même temps que ces nouvelles technologies.

1. Stoll, C. (1988). Stalking the wily hacker. Communications of the ACM, 31(5), 484-497.

2. [https://en.wikipedia.org/wiki/The_Cuckoo%27s_Egg_\(book\)](https://en.wikipedia.org/wiki/The_Cuckoo%27s_Egg_(book))

1 Introduction

Les cyberattaques n'épargnent personne. Elles peuvent aussi bien toucher les particuliers que des états, tout en passant par des organisations de toutes tailles. Les objectifs des attaques varient également. Un particulier pourrait par exemple être victime d'un mail de phishing dont l'objectif serait de lui extorquer de l'argent. Un groupe d'attaquants pourrait également viser une organisation avec un *ransomware* afin de tenter de récupérer une rançon en échange d'une clé de déchiffrement. Mais l'aspect pécuniaire n'est pas toujours la finalité. En effet, les cyberattaques peuvent également poursuivre des objectifs géopolitiques, pour tenter de déstabiliser un état ennemi, comme cela a été le cas en Ukraine avec le *malware NotPetya*³ ou le cas de l'ingérence russe durant les élections aux États-Unis⁴.

De plus, le nombre d'attaques tend à augmenter dans le temps. En 2021, en pleine pandémie, on estime leur nombre à presque 20 millions dans le monde⁵. Ces attaques ont également un coût considérable. On évalue par exemple qu'en 2024 le coût de la cybercriminalité a atteint les 9 000 milliards de dollars américains et devrait continuer d'augmenter dans le temps⁶. Mais l'impact d'une attaque n'est pas toujours quantifiable. En 2016, par exemple, un logiciel espion développé par la société *NSO* a été découvert sur le téléphone de plusieurs personnalités politiques, des membres d'ONG ou des journalistes. Celui-ci étant vendu à des États ou gouvernements afin d'espionner des opposants⁷. En 2010, *Stuxnet* a été découvert et analysé par des chercheurs qui l'ont défini comme étant un des logiciels malveillants les plus complexes jamais conçu. Celui-ci aurait été développé par la NSA et l'unité israélienne *Unit 8200* et a permis de détruire des centrifugeuses de centrales nucléaires iraniennes afin d'affaiblir le régime en place⁸. Il est difficile de mesurer l'effet de ce type de pratique.

Face aux défis qu'imposent ces cyberattaques, il est apparu évident aux experts que renforcer la sécurité d'une infrastructure ne serait pas suffisant. Il faut également se préparer à l'éventualité que, malgré les mesures mises en place, une faille soit exploitée et un système informatique soit compromis. Cette capacité à réagir a pris la forme d'une compétence en soi : *la réponse d'incident*, généralement prise en charge par une équipe appelée *CSIRT (Computer Security Incident Response Team)*.

De plus, afin de protéger les citoyens qui utilisent les nouvelles technologies mais également pour renforcer la sécurité des infrastructures critiques des états, certaines normes légales ont été mises en place. Le règlement général sur la protection des données (RGPD) est un bon exemple d'application concrète de la loi afin de protéger les données personnelles des utilisateurs. Ce règlement mentionne par exemple qu'en cas de cyberincident, les autorités de contrôle et les personnes concernées par l'incident doivent être informées, notamment pour définir l'ampleur de la compromission des données⁹. La directive européenne NIS2 vise elle à identifier les infrastructures dites critiques en Europe afin de leur imposer une série de mesures à respecter en termes de cybersécurité. L'article 21 de cette directive met par exemple en avant l'importance de

3. https://fr.wikipedia.org/wiki/Cyberattaque_NotPetya

4. https://en.wikipedia.org/wiki/Russian_interference_in_the_2016_United_States_elections

5. <https://www.statista.com/forecasts/1485031/cyberattacks-annual-worldwide/>

6. <https://www.statista.com/chart/28878/expected-cost-of-cybercrime-until-2027/>

7. [https://fr.wikipedia.org/wiki/Pegasus_\(logiciel_espion\)](https://fr.wikipedia.org/wiki/Pegasus_(logiciel_espion))

8. <https://fr.wikipedia.org/wiki/Stuxnet>

9. Il s'agit des articles 33 et 34 du règlement

mettre en place des processus de réponse d'incidents¹⁰. Afin de répondre à ces exigences il est essentiel de mettre en place des stratégies pour faire face à des cyberattaques mais également de pouvoir les comprendre et de caractériser leurs impacts. Ces règlements et directives soulignent par exemple l'importance de notifier les victimes et les instances en charge afin de leur exposer l'impact d'un incident. Il sera pour cela nécessaire de faire appel à des analystes spécialisés qui seront en mesure d'identifier les artefacts laissés par des attaquants, de comprendre les techniques utilisées, de poser des hypothèses sur les objectifs poursuivis et de définir la nature et la quantité de données ayant été compromises. Ce travail nécessite d'appliquer une méthodologie précise afin d'assurer la pertinence des preuves numériques récoltées et d'assurer la recevabilité de celles-ci devant des instances judiciaires si cela est nécessaire. Face à un contexte qui est différent d'un incident et d'une victime à l'autre, les analystes vont devoir se poser les questions suivantes : quelles preuves récolter en priorité et pourquoi? Comment les collecter? Quels outils existent pour effectuer cela tout en conservant la validité des artefacts identifiés? Comment interpréter ces preuves numériques et comment effectuer une corrélation entre celles-ci? C'est pour vous aider à répondre à ces questions que nous avons écrit ce livre.

1 À qui s'adresse ce livre

Il y a quelques années, lorsque nous avons dû mettre en place un cours de réponse d'incident et d'analyse forensique pour l'enseignement supérieur, il a été difficile d'identifier une ressource qui synthétisait cette thématique. En effet, il existe de nombreuses ressources qui traitent de matières spécialisées comme celle de la rétro-ingénierie¹¹, de l'analyse de logiciels malveillants¹², de la capture d'image disque¹³ ou encore de l'analyse de mémoire vive¹⁴. D'autres ouvrages traitent de la réponse d'incident et de la forensique mais en survolant les aspects techniques ou en se limitant à citer les différents outils existants sans vraiment rentrer dans les détails de leur fonctionnement. Pour finir, la plupart des ouvrages existants sont disponibles en anglais. Notre premier objectif a donc été d'écrire un livre qui permettrait à des apprenants de bien comprendre les fondements techniques qui constituent une réponse d'incident, en justifiant le choix des artefacts à récolter tout en expliquant le fonctionnement des outils utilisés. En effet, le contexte d'un incident pouvant fortement changer d'une victime à l'autre, il nous a semblé important de revenir sur les bases du fonctionnement d'une infrastructure informatique afin que le lecteur puisse adapter ses connaissances aux incidents qu'il pourrait rencontrer. Des connaissances de base en informatique et en sécurité sont toutefois nécessaires pour mieux appréhender cet ouvrage car il n'est évidemment pas possible de revenir sur les concepts généraux des sciences informatiques auxquelles nous faisons référence tout au long des chapitres.

10. <https://eur-lex.europa.eu/legal-content/FR/TXT/HTML/?uri=CELEX:32022L2555>

11. <https://beginners.re/>

12. Practical Malware Analysis : The Hands-On Guide to Dissecting Malicious Software 1st Edition, Michael Sikorski, Andrew Honig

13. Practical Forensic Imaging : Securing Digital Evidence with Linux Tools 1st Edition, Bruce Nikkel

14. The Art of Memory Forensics : Detecting Malware and Threats in Windows, Linux, and Mac Memory 1st Edition, Michael Hale Ligh, Andrew Case, Jamie Levy, Aaron Walters

1 Introduction

Ce livre s'adresse donc aussi bien à des enseignants ou formateurs qui souhaiteraient s'appuyer sur un ouvrage de référence pour construire leur cours, qu'à des professionnels qui voudraient compléter leurs connaissances autour de ces thématiques. Nous avons donc tenté de trouver un juste équilibre entre vulgarisation et profondeur technique afin d'apporter une réelle valeur ajoutée à cet ouvrage.

2 Les thématiques abordées dans ce livre

Les thèmes de la cybersécurité qui orbitent autour d'une réponse d'incident sont nombreux. Nous n'avons évidemment pas pu traiter ou creuser toutes ces thématiques dans cet ouvrage. C'est pour cela qu'il est important pour nous de définir clairement notre périmètre d'étude. Premièrement, nous nous sommes limités à l'étude des systèmes d'exploitation Windows et Linux. Ceux-ci étant respectivement les plus touchés comme poste de travail et comme serveur. Pour chacun de ces systèmes, nous allons identifier et justifier les différents artefacts qu'il est important d'analyser après un incident. Nous montrerons ensuite comment récolter ces artefacts et quel impact le choix de la technique peut avoir sur la pertinence des résultats. Des langages de scripting seront utilisés comme le *Bash* sous Linux ou *Powershell* pour Windows. Nous verrons aussi comment faciliter cette récolte à l'aide d'outils comme *Velociraptor*. Concernant la capture de mémoire vive et d'image disque, nous aborderons les enjeux liés à ces captures tout en présentant des outils comme : *FTK Imager* et *AVML*. Pour la partie analyse, différentes technologies seront utilisées mais nous discuterons plus particulièrement de la suite *Sleuth Kit* et de l'outil *Volatility3*. Pour finir, un chapitre s'attardera plus spécifiquement sur l'analyse de logiciels malveillants.

3 Les sujets qui ne seront pas discutés dans ce livre

Comme nous le verrons tout au long de cet ouvrage, le nombre et la qualité des preuves récoltées dépend fortement de l'infrastructure existante. La mise en place d'un système de défense et de surveillance qui facilite une réponse d'incident est un sujet en soi et il ne sera pas traité ici. La description des indicateurs de compromission permettra néanmoins au lecteur d'avoir un bon aperçu de ce qui distingue une infrastructure configurée pour fournir les preuves nécessaires au moment d'une attaque, de celles qui ne le sont pas.

En amont de la mise en place de mesures techniques pour se préparer à un incident, il est nécessaire de définir une stratégie générale de sécurité. Ces aspects de gouvernance qui touchent également la réponse d'incident ne seront pas traités dans ce livre car elles dépassent largement le périmètre défini.

Afin de compléter une enquête numérique il est parfois nécessaire de corréler les indicateurs de compromission et des techniques identifiées avec des données disponibles gratuitement sur Internet. Ces disciplines sont appelées *OSINT (Open Source Intelligence)* et *Threat Intelligence*. Même si nous aborderons ces aspects dans notre ouvrage, d'autres publications plus spécialisées traitent de ces thématiques plus en profondeur¹⁵.

15. Intelligence-Driven Incident Response : Outwitting the Adversary 2nd Edition, Rebekah Brown, Scott J. Roberts

Pour finir, comme discuté plus tôt, les systèmes d'exploitation macOS¹⁶, iOS et Android¹⁷ ne seront pas étudiés ici. Il existe néanmoins des ouvrages qui s'intéressent plus particulièrement à ces systèmes et qui permettront au lecteur de compléter ses connaissances si nécessaire.

4 Organisation du livre

Le livre s'articule en trois parties. Dans la première partie, "Indicateurs de compromission", nous répondrons à la question suivante : "Quels éléments récolter et pourquoi?" Nous verrons dans cette partie où trouver les différents indicateurs de compromission pour les systèmes Windows et Linux. Nous montrerons également comment distinguer des signes d'un fonctionnement normal pour un système de ceux qui peuvent être considérés comme suspects. La deuxième partie appelée "Récolte de preuves numériques" permettra de répondre à la question suivante : "Comment récolter les artefacts sur un système tout en conservant la validité des preuves". Nous présenterons différents outils afin d'effectuer cette récolte tout en exposant les principaux enjeux autour de la capture et du respect d'une méthodologie forensique. La troisième et dernière partie de ce livre, "Analyse d'un incident", présentera l'application des différentes techniques et outils définis dans les parties 1 et 2 de ce livre à un incident inspiré d'un cas réel. En effet, nous lancerons une attaque sur un système virtualisé afin d'illustrer les concepts étudiés dans ce livre. Pour chaque partie nous proposerons également une série d'exercices et nous présenterons des ouvrages qui permettront aux lecteurs de compléter leurs connaissances dans une thématique qui les intéresse plus particulièrement.

5 Supports pédagogiques

L'ensemble des ressources pédagogiques nécessaires pour reproduire les différentes manipulations illustrées dans ce livre ainsi que les exercices seront disponibles sur le dépôt *github* suivant : https://github.com/voisina/la_reponse_incident. Vous pouvez également scanner le QR code ci-dessous :



FIGURE 1.1 – QR Code vers les ressources du livre.

16. The Art of Mac Malware, Volume 2, Detecting Malicious Software, Patrick Wardle

17. Android Security Internals : An In-Depth Guide to Android's Security Architecture, Nikolay Elenkov

Première partie

Indicateurs de compromission

2 Introduction

Lors de ses enquêtes, la police scientifique s'appuie sur le postulat qu'un suspect aura forcément laissé des traces en réalisant son acte malveillant. C'est le fameux principe d'échange de Locard¹. La réussite de l'enquête reposera donc sur la quantité et la qualité de ces traces qui prendront éventuellement la forme de preuves. Le même principe s'applique à l'investigation numérique où les données capturées et récoltées permettront de mieux comprendre un incident et de proposer une réponse adéquate. Et comme pour une enquête de police, la méthode de récolte et de préservation des preuves est essentielle afin de garantir la pertinence de celles-ci. Dans le contexte d'infrastructures informatiques ces traces sont appelées des "indicateurs de compromission" et peuvent prendre différentes formes. Il pourra par exemple s'agir d'une communication réseau inhabituelle, des journaux d'un serveur qui mettent en évidence une attaque sur un mot de passe ou encore des processus suspects sur un poste de travail en particulier. Il est dès lors évident qu'identifier ces indicateurs dépendra fortement de la taille et de la structure de l'infrastructure visée ainsi que de la maturité en termes de cybersurveillance de l'entreprise se reposant sur cette infrastructure. De plus, afin de rendre une réponse d'incident efficace, il est nécessaire de s'appuyer sur une méthodologie et une série d'outils permettant d'effectuer des vérifications systématiques tout en étant capable de s'adapter aux particularités de la victime.

L'enjeu pour la récolte d'information est donc triple. Premièrement, en amont d'un potentiel incident, il faut s'assurer que d'éventuelles preuves soient bien capturées et utilisables en cas d'attaque ou de panne. C'est aux administrateurs d'une infrastructure de déterminer le niveau de journalisation d'un système et de son taux de rétention². Deuxièmement, en fonction du contexte d'un incident, il faudra déterminer quelles informations seront importantes à récolter en priorité. En effet, la masse de données échangées et stockées dans une infrastructure est considérable et analyser l'ensemble de celles-ci n'est pas réalisable en pratique. Pour terminer, il sera nécessaire d'identifier la meilleure méthode en fonction du contexte pour collecter les preuves. La méthode dépendra notamment de la taille de l'infrastructure, de sa localisation (hébergée localement ou dans le cloud) et des outils mis en place par l'entreprise pour faciliter cette collecte.

Dans ce chapitre, nous définirons et justifierons quels sont les artefacts importants à identifier et à capturer dans le contexte d'une réponse d'incident sur des systèmes Windows et Linux. Dans la partie suivante, nous nous intéresserons plus particulièrement aux outils et techniques existants pour faciliter, structurer et adapter cette récolte.

1. https://fr.wikipedia.org/wiki/Principe_d'%C3%A9change_de_Locard

2. Durée durant laquelle un journal sera conservé sur un disque

3 Indicateurs de compromission

Windows

Windows se différencie de Linux par le fait qu'il s'agit d'un système propriétaire pour lequel il n'est pas toujours possible d'obtenir une documentation technique précise. Cela rend le travail d'analyse parfois plus compliqué dans le sens où il est souvent nécessaire de vérifier une information à plusieurs emplacements et d'utiliser des logiciels tiers capables de décoder les différents formats utilisés par Microsoft dans son système d'exploitation. Nous avons néanmoins réalisé une synthèse des composants qui nous semblent essentiels à analyser afin d'en extraire des preuves numériques dans le contexte d'un incident.

1 Le registre Windows

Le registre Windows est une base de données hiérarchique qui permet de paramétrer une partie du fonctionnement du système d'exploitation. De nombreux composants de Windows ainsi que des applications tierces vont interagir avec ce registre et dès lors, y laisser des traces de leur activité. Pour un analyste, il s'agit d'une source d'information riche pour tenter de reconstituer la ligne du temps d'une attaque. De plus, et comme nous le verrons dans le chapitre dédié à l'analyse de logiciels malveillants, les données contenues dans ce registre peuvent être modifiées par un attaquant ou par un de ses outils pour garantir une persistance ou élever ses privilèges. Il est donc important de vérifier certaines valeurs stockées dans cette base de données afin de s'assurer que rien d'anormal n'est détecté.

Le registre Windows est présent sur le disque sous la forme d'un ensemble de fichiers dans un format binaire propriétaire et se trouve à l'emplacement suivant :

```
1 C:\Windows\System32\config
```

Nous nous intéresserons tout particulièrement aux clés de registre contenues dans les fichiers suivants :

- ▶ **SAM** : contient des informations sur les utilisateurs et les groupes.
- ▶ **SECURITY** : enregistre des paramètres liés aux politiques locales de sécurité et aux authentifications.
- ▶ **SOFTWARE** : contient les données utilisées par les différentes applications installées sur le système.
- ▶ **SYSTEM** : stocke essentiellement les paramètres utilisés par les pilotes et le système de démarrage.

Ces fichiers binaires sont appelés des "*hives*" et seront, soit chargés en mémoire au démarrage du système, soit à la demande en fonction du contexte d'exécution, ce qui permet de constituer la base de données que l'on appelle registre Windows.

Attention que certains registres sont stockés à d'autres emplacements, comme c'est par exemple le cas du "*hive*" spécifique à chaque utilisateur, *NTUSER.DAT*, qui lui est stocké à l'emplacement suivant :

```
1 C:\Users\USERNAME
```

Pour finir, un autre *hive* que nous étudierons ici est l'*Amcache.hve*, composant du système de compatibilité des applications de Windows. Ce dernier est stocké ici :

1 C:\Windows\appcompat\Programs

L'ensemble des *hives* permet donc de constituer le registre qui est composé d'un ensemble de clés organisées de manière hiérarchique. Chaque clé peut contenir des valeurs, enregistrées selon différents types¹. Voici les principaux types étudié ici :

- ▶ REG_SZ et REG_EXPAND_SZ : Chaîne de caractères, simple ou étendue (permettant d'utiliser des variables d'environnement)
- ▶ REG_BINARY : Données stockées en binaire
- ▶ REG_DWORD et REG_QWORD : Nombre, stocké respectivement en 32 ou 64 bits
- ▶ REG_LINK : Lien vers une autre clé de registre

Afin de parcourir les différentes clés sur un système en activité, il est possible d'utiliser l'application Windows prévue à cet effet : *Registry Editor*. L'image 3.1 illustre l'organisation du registre en clés hiérarchiques.

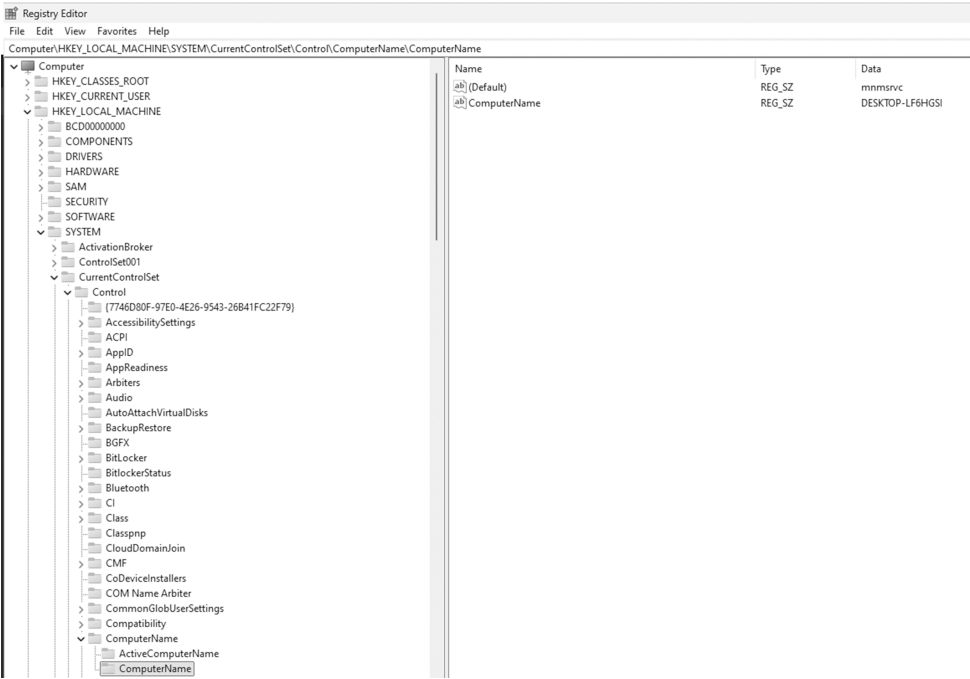


FIGURE 3.1 – Visualisation du registre à l'aide de *Registry Editor*.

Dans cet exemple, nous recherchons une clé de registre en particulier, "*ComputerName*" en indiquant dans la barre de recherche en haut de l'écran, le chemin complet vers celle-ci. Le panneau de gauche montre que cette clé est contenue dans la clé racine "*LOCAL_MACHINE*" car ce paramètre est bien lié à la machine et non à un utilisateur. La valeur est ensuite stockée dans une clé avec le type REG_SZ, c'est-à-dire une chaîne

1. <https://learn.microsoft.com/en-us/windows/win32/sysinfo/registry-value-types>

de caractères correspondant au nom donné à la machine. Dans le contexte d'une analyse forensique, il sera souvent désirable de travailler à partir d'une capture de l'image disque. Les fichiers *hive* pourront dès lors être exportés et chargés dans un autre logiciel comme *Registry Explorer*² qui permettra à la fois de visualiser les clés de manière hiérarchique, mais également d'interpréter les valeurs stockées le plus souvent au format binaire. Un exemple d'utilisation de cet outil sera présenté plus tard dans ce chapitre.

L'extrait 3.1 reprend une liste non exhaustive des clés de registre particulièrement intéressantes dans le contexte d'une analyse forensique.

```

1  % Nom de la machine et numéro de version
2  HKEY_LOCAL_MACHINE\SYSTEM\
3  CurrentControlSet\Control\ComputerName\ComputerName
4  HKEY_LOCAL_MACHINE\SOFTWARE\
5  Microsoft\Windows NT\CurrentVersion
6
7  % Programmes installés sur la machine
8  HKEY_LOCAL_MACHINE\SOFTWARE\
9  Microsoft\Windows\CurrentVersion\Uninstall
10
11 % Liste des programmes lancés au démarrage
12 HKEY_LOCAL_MACHINE\Software\
13 Microsoft\Windows\CurrentVersion\Run
14 HKEY_LOCAL_MACHINE\Software\
15 Microsoft\Windows\CurrentVersion\RunOnce
16 HKEY_LOCAL_MACHINE\SYSTEM\
17 CurrentControlSet\Services
18 HKEY_LOCAL_MACHINE\SOFTWARE\
19 Microsoft\WindowsNT\CurrentVersion\Schedule\TaskCache
20
21 % Accès récents aux fichiers
22 HKEY_CURRENT_USER\Software\
23 Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
24 HKEY_LOCAL_MACHINE\SYSTEM\
25 CurrentControlSet\Control\Session Manager\AppCompatCache
26 HKEY_CURRENT_USER\Software\
27 Microsoft\Windows\CurrentVersion\Explorer\UserAssist
28
29 % Supports amovibles et disques
30 HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices
31 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR
32
33 % Shellbags
34 HKEY_CURRENT_USER\Software\
35 Classes\LocalSettings\Software\Microsoft\Windows\Shell\

```

Extrait 3.1 – Liste non exhaustive de clés de registres intéressantes lors d'une analyse forensique.

Les deux premières clés de registre donnent des indications sur le nom de la machine visée ainsi que sur les versions du système d'exploitation. Avec ces informations, unana-

2. <https://ericzimmerman.github.io/>

lyste peut structurer son investigation en identifiant les machines et en vérifiant si les versions des systèmes sont potentiellement vulnérables à certaines attaques.

La clé de registre suivante permet elle, d'obtenir des informations concernant les logiciels installés sur la machine cible. Cette information est particulièrement utile si l'on veut par exemple s'assurer qu'un logiciel vulnérable ou malveillant est installé ou non sur un poste de travail.

Comme nous le verrons dans la partie de ce livre dédié à l'analyse de *malwares*, ceux-ci peuvent modifier des clés de registre afin de rendre leur exécution persistante à un redémarrage. Les quatre clés de registre mentionnées ici permettent de vérifier la liste des tâches et services lancés au démarrage³.

Il est également important de comprendre les agissements d'un utilisateur juste avant une analyse. Les clés *RecentDocs*, *AppCompatCache* et *UserAssist* permettent de poser des hypothèses sur les fichiers récemment ouverts ainsi que les programmes potentiellement exécutés dernièrement sur la machine. Nous verrons également dans ce chapitre comment corréler ces données avec d'autres informations générées par des composantes de Windows.

On retrouve ensuite deux clés de registre qui peuvent être consultées afin de vérifier si des supports amovibles ont été connectés à une machine. Cela peut s'avérer particulièrement utile si on suspecte une propagation de logiciels malveillants via ce type de vecteur. Par exemple, il est possible d'identifier la date à laquelle une clé USB a été connectée ou déconnectée en consultant les propriétés dans le registre "USBSTOR/Nom De l'appareil USB/ID/Properties". L'identifiant 0064 donne la date de première connexion et les identifiants 0066 et 0067 donnent respectivement les dates de la dernière connexion et de la dernière déconnexion.

La dernière clé de registre mentionnée ici fait référence au système de *shellbag* sous Windows. Cette famille de clés est utilisée pour enregistrer les préférences dans la visualisation des informations dans Windows Explorer. Ce faisant, le système stocke un nombre important d'informations sur les interactions d'un utilisateur avec les différents dossiers et fichiers présents ou supprimés du système.

À noter que toutes les données présentes dans le registre Windows ne sont pas exploitables "en l'état". Si certaines informations comme le nom de la machine ou sa version sont lisibles directement dans les paramètres de la clé concernée, d'autres nécessiteront de passer par un outil tiers afin de les interpréter. C'est par exemple le cas du registre *AppCompatCache* qui pourra être rendu lisible à l'aide d'outils comme *ShimCacheParser*⁴. Un outil d'analyse d'image disque comme *Autopsy* permet d'harmoniser et de structurer les différentes informations récupérées dans le registre. Nous verrons comment utiliser *Autopsy* dans le chapitre dédié à l'analyse de preuves numériques.

2 Event Logs

La journalisation des systèmes informatiques est une composante essentielle qui permet aux spécialistes de détecter et de mieux comprendre des anomalies dans le

3. Cette liste n'est pas exhaustive, il en existe d'autres, moins fréquemment utilisées

4. <https://github.com/mandiant/ShimCacheParser>

fonctionnement des différentes applications qui constituent ce système. Les journaux⁵ sont également une source d'information très riche pour des analystes, car, si ceux-ci sont bien paramétrés, alors ils peuvent contenir les traces laissées par un attaquant. Les systèmes Windows modernes utilisent le gestionnaire de journalisation propriétaire *Windows Event Logs*, dont les fichiers sont stockés au format *.evtx*. Ces fichiers sont divisés en différentes catégories dont les principales sont : *Application*, *Security* et *System*. Le tableau 3.1 résume les informations que l'on peut retrouver par catégorie.

Catégorie de journal	Objectif	Exemple
System	Enregistre les événements liés au fonctionnement du système d'exploitation	Erreurs matérielles, démarrage et arrêt
Application	Enregistre les événements liés aux applications qui s'exécutent sur le système	Plantage d'une application en cours d'exécution
Security	Enregistre les événements liés à la sécurité du système	Modification des utilisateurs et des permissions. Échec d'authentification
Setup	Enregistre les événements liés aux installations des mises à jour ou des applications	Mise à jour Windows, installation de fonctionnalités

TABLE 3.1 – Informations utiles par catégories.

À noter qu'il existe également des journaux plus spécifiques au fonctionnement des services et des applications. On y retrouve par exemple les événements liés au serveur *DNS*, à un *Active Directory* ou encore au système de planificateur de tâches. Chaque événement dans un journal se voit attribuer un "niveau" de gravité. On retrouve par ordre croissant d'importance : *Information*, *Warning*, *Error*, *Critical*. Ces niveaux de gravité seront particulièrement utiles pour résoudre un problème technique, mais c'est bien la corrélation entre la gravité, le type et le nombre d'occurrences d'un événement sur lequel un analyste sera attentif pour déterminer la nature et l'ampleur d'une possible cyberattaque. La catégorie de journaux de type *Security* est enrichie par le résultat d'un audit qui pourra prendre deux valeurs : *Success* ou *Failure*. Par exemple, si un utilisateur se trompe de mot de passe ou si un accès à un fichier lui est refusé, l'entrée de l'évènement sera indiquée comme "*Audit Failure*". Filtrer ces entrées sur base de ce statut permettra de potentiellement mettre en évidence des tentatives d'élévation de privilèges ou d'attaque sur les mots de passe.

Afin de faciliter le tri, chaque type d'évènement se distingue par un identifiant unique. Un analyste pourra dès lors constituer une sélection de ceux-ci pour les collecter en priorité sur base de cet identifiant. Le tableau 3.2 présente une liste non exhaustive d'identifiants de journaux à vérifier dans le contexte d'une réponse d'incident.

Consulter les journaux liés à l'authentification sur un ensemble de machines permet de vérifier si des attaques sur les mots de passe n'ont pas été effectuées. En effet, lorsqu'un attaquant veut se connecter à une machine ou à un service, il peut tenter d'entrer une série de mots de passe courants pour un utilisateur donné en espérant tomber sur

5. Plus couramment appelés "*logs*" en anglais

Event ID	Catégorie	Description
4625	Security	Échec d'authentification
4670	Security	Les permissions sur un objet ont été modifiées
4663	Security	Tentative d'accès à un fichier ou un objet
4688, 4689	Security	Nouveau processus créé, arrêté
4697	Security	Un service a été installé
4698	Security	Nouvelle tâche de planification créé
4720	Security	Un nouvel utilisateur a été créé
4672	Security	Authentification avec des privilèges élevés
1102	Security	Les journaux audit ont été supprimés
6005, 6006	Security	Le service d'audit a été démarré, arrêté

TABLE 3.2 – Liste d'identifiants de journaux à vérifier.

la bonne combinaison. Ce type d'attaque est appelée "attaque par force brute" ou encore "attaque dictionnaire". Cette technique est toutefois considérée comme "bruyante" dans le sens où cela risque de rapidement déclencher des alertes sur des systèmes de surveillance, de bloquer des comptes ou de créer un nombre important d'événements dans les journaux Windows. De plus, si les mots de passe sont considérés comme suffisamment "forts", alors ce type d'attaque ne fonctionnera pas⁶. Malgré tout, cette technique reste fortement utilisée et il est crucial de vérifier les événements qui en découlent. De plus, même si l'attaquant ne tente pas de "casser" des mots de passe, connaître les dates et heures auxquelles un utilisateur s'est connecté ou déconnecté d'une machine permettra potentiellement d'enrichir l'analyse de la chronologie d'un incident.

D'autres journaux à vérifier sont ceux qui enregistrent des activités liées aux permissions sur des ressources comme des fichiers ou des applications. Un attaquant va généralement tenter d'accéder à des ressources critiques. Il n'a pas toujours directement accès à ces données et il devra alors effectuer une élévation de privilèges pour arriver à ses fins. Si un utilisateur se voit refuser l'accès à une ressource cela déclenchera un événement dans les journaux au même titre que s'il tente de modifier les permissions sur celle-ci.

Les événements 4688, 4689, 4697 et 4698 permettent de vérifier l'activité d'un événement logiciel malveillant. En effet ceux-ci vont traditionnellement lancer de nouveaux processus et tenter de se rendre persistant à l'aide de services ou de tâches de planification. Les actions réalisées sur les utilisateurs doivent également être surveillées en priorité. Il est courant qu'un attaquant crée ou modifie les permissions d'un utilisateur ou qu'il tente de l'ajouter dans un groupe particulier afin d'effectuer des déplacements latéraux dans l'infrastructure. Nous verrons cette partie plus en détails dans le contexte des journaux liés à l'*Active Directory*.

Les événements portant les identifiants : 1102, 6005 et 6006 permettent de s'assurer que la journalisation de Windows était bien opérationnelle durant un incident dans le but de valider la pertinence des éléments collectés.

Aujourd'hui la plupart des infrastructures informatiques reposent sur des annuaires

6. Pour être plus précis, une attaque par force brute finira par trouver la bonne combinaison. Toutefois, le délai de ce type d'attaque dépend fortement de la force du mot de passe, c'est-à-dire de sa complexité et de sa longueur. Si un mot de passe n'est pas présent dans un dictionnaire et si sa longueur est suffisamment importante, alors la durée pour trouver celui-ci à l'aide d'une attaque par force brute la rend irréaliste dans la pratique.

pour centraliser les accès aux ressources, comme l'*Active Directory* de Microsoft⁷. Ce système permet de réunir les utilisateurs, les ordinateurs et les différentes ressources au sein d'un domaine, tout en définissant qui a accès à quoi. Un utilisateur ne se connecte donc plus localement sur sa machine, mais dans le domaine et les permissions lui sont accordées par un serveur central. Ce système d'authentification et de distribution des accès repose sur le protocole *Kerberos*. Sans rentrer dans les détails, ce protocole est composé de trois entités, le client (qui désire accéder à des ressources), un serveur central (appelé contrôleur de domaine dans le contexte de l'AD Microsoft) et d'une série de ressources auxquelles un utilisateur peut accéder. Pour simplifier, un utilisateur s'authentifie sur le serveur central qui pourra lui générer des tickets utilisables auprès des ressources auxquelles il voudrait accéder. Tout ce système d'authentification produit également des événements qui seront enregistrés dans les journaux du serveur qui joue le rôle de contrôleur de domaine. Le tableau 3.3 présente un résumé des identifiants d'événements à surveiller dans le contexte d'une infrastructure gérée au sein d'un Active Directory.

Event ID	Catégorie	Description
4771	Security - Audit Failure	Échec d'une pré-authentification Kerberos
4776	Security - Audit Failure	Le contrôleur de domaine a tenté de valider les informations d'identification d'un compte
4798, 4799	Security - Audit Success	Énumération des appartenances aux groupes
5136, 5137, 5141	Security - Audit Success	Objet de l'AD modifié
4723	Security - Audit Success	Modification d'un mot de passe
4732	Security - Audit Success	Ajout d'un utilisateur dans un groupe
4735	Security - Audit Success	Modification des propriétés d'un groupe

TABLE 3.3 – Résumé des identifiants à surveiller dans le contexte d'une infrastructure reposant sur un AD.

Concernant les tentatives d'authentification (réussies ou échouées), le contrôleur de domaine partage les mêmes identifiants décrits précédemment. Malgré tout, ceux-ci peuvent être enrichis par les événements 4771 et 4776. Le premier indique une tentative échouée pour la première étape d'une authentification *Kerberos*, ce qui indiquera qu'un utilisateur qui tente de se connecter à un compte lié au domaine a renseigné de mauvais identifiants. Le deuxième indique une authentification à l'aide du protocole *NTLM*. Son utilisation dans le contexte d'un domaine peut potentiellement indiquer une attaque de type *Pass The Hash*⁸.

Afin d'effectuer des déplacements latéraux dans une infrastructure, un attaquant peut énumérer les différents utilisateurs, groupes et objets liés à un domaine pour mieux comprendre son fonctionnement et y déceler d'éventuelles failles. Des outils comme *Bloodhound* permettent d'automatiser ce processus. Ce genre d'outil effectuera un nombre important de requêtes vers le contrôleur de domaine afin de construire sa base

7. À noter qu'il existe l'équivalent de ce genre de système directement hébergé dans le *cloud*, comme *Microsoft Entra ID*. Dans le contexte de ce livre, nous nous limiterons à l'analyse d'indicateurs de compromission dans des solutions hébergées localement dans l'entreprise.

8. https://en.wikipedia.org/wiki/Pass_the_hash

de données. Ces requêtes laissent des traces et il est par exemple possible de consulter les journaux 4798 et 4799 pour vérifier si une tentative d'énumération a été effectuée.

Une fois la reconnaissance sur le domaine terminée, un attaquant va tenter d'exploiter une possible vulnérabilité dans le but d'élever ses privilèges ou de se connecter à d'autres machines. Durant ce processus il devra parfois modifier des permissions, créer de nouveaux utilisateurs ou les placer dans des groupes de sécurité bénéficiant de permissions spéciales, etc. Cela dépendra fortement du contexte, mais toutes ces actions laissent des traces dans les journaux. Les événements de type 5136, 5137 et 5141 alertent par exemple sur la modification, la création ou la suppression d'un objet dans le domaine (un utilisateur, un groupe, une machine, etc). D'autres événements permettent de surveiller des changements plus précis comme l'ajout d'un utilisateur dans un groupe (4732) ou la modification d'une propriété d'un groupe (4735).

Les fichiers d'événements se trouvent généralement dans le répertoire suivant pour une machine Windows récente :

```
1 C:\Windows\System32\winevt\Logs
```

Ces fichiers sont enregistrés dans un format binaire propriétaire avec l'extension *.evtx*. Il n'est pas possible dès lors de les lire tel quel. Il faut passer par une application qui permet de les interpréter ou de les convertir. Windows fournit un outil de visualisation appelé *Event Viewer* qui permet de naviguer et filtrer les différents événements. Même si cet outil peut s'avérer pratique pour comprendre une erreur sur un système, il n'est par contre pas le plus approprié pour identifier une attaque ou des indicateurs de compromission dans le contexte d'une réponse d'incident. Premièrement, il faut consulter l'ensemble des fichiers pour une infrastructure qui peut potentiellement être composée de centaines de postes de travail. De plus, il est utile de coupler la lecture de ces événements avec d'autres outils qui permettent d'identifier des menaces plus rapidement en effectuant des corrélations ou en mettant en évidence certains journaux critiques en fonction du contexte.

Il existe des applications qui permettent de décoder et d'extraire les journaux Windows dans un format CSV⁹. Une fois extrait sous cette forme, il est alors possible d'utiliser un langage de script afin de trier le contenu et d'identifier de l'activité malveillante en fonction du contexte de l'analyse. Un autre outil, *chainsaw*¹⁰, permet, quant à lui, de confronter un journal *evtx* avec des règles *Sigma*¹¹ pour mettre en évidence des techniques d'attaques connues. Attention toutefois qu'une attaque ne s'appuie pas toujours sur des techniques ou des outils connus. Dans ce cas-là, l'utilisation d'un outil de triage comme *chainsaw* peut s'avérer insuffisant pour l'analyse des journaux Windows.

Notons toutefois que tous les journaux discutés ici ne sont pas forcément activés par défaut sur des postes Windows. Il est possible de gérer la politique de journalisation soit dans la politique locale de sécurité de Windows ou à l'aide des *GPO* dans le contexte de machines appartenant à un domaine *Active Directory*.

9. <https://github.com/EricZimmerman/evtx>

10. <https://github.com/WithSecureLabs/chainsaw>

11. <https://github.com/SigmaHQ/sigma>

Yara Sigma Et cetera

Analyser des preuves numériques nécessite souvent de traiter une masse importante d'informations, comme des journaux systèmes, des captures de trafic réseau ou des fichiers binaires. Afin de faciliter le travail des enquêteurs, il est possible d'utiliser des règles qui fonctionnent sur la base de reconnaissance d'une "empreinte" type dans un fichier. Il existe différents formats utilisés par différentes applications. *Yara*^a étant un langage de règles fréquemment utilisé pour analyser des fichiers potentiellement malveillants. Des IDS comme *Snort*^b ou *Suricata*^c vont utiliser un autre format afin de détecter du trafic réseau suspect. Les règles *Sigma* poursuivent elles l'objectif d'offrir un format générique afin d'analyser des journaux provenant de différentes sources. Dans tous les cas, le principe reste le même, à savoir documenter des attaques et des techniques en écrivant des règles permettant de les détecter à l'avenir. Un défenseur pourra alors ajouter ces règles dans ses différents outils d'analyse afin de pouvoir rapidement détecter une attaque ou une tentative d'intrusion dans son infrastructure.

a. <https://yara.readthedocs.io/en/stable/writingrules.html>

b. <https://www.snort.org/>

c. <https://suricata.io/>

3 Processus et services

Rien ne représente mieux l'état d'une machine que la liste et l'état des processus et services en fonctionnement sur celle-ci. Lorsqu'un programme est exécuté, un (ou plusieurs) processus sont créés. Cela permet de gérer l'utilisation de la mémoire du programme ainsi que les permissions desquelles il pourra bénéficier. Ce fonctionnement permet de fournir aux utilisateurs du multi-tâche, c'est-à-dire de pouvoir exécuter en parallèle une multitude d'applications. Les services permettent d'exécuter un programme en arrière-plan, mais également d'apporter d'autres fonctionnalités, comme, par exemple, automatiser l'exécution de ces programmes au démarrage. Ceux-ci sont pour la plupart utilisés par le système d'exploitation ou par les serveurs. En d'autres termes, consulter la liste des processus et des services permet de déterminer ce qu'il se passe sur une machine à un instant t , ce qui est primordial dans le contexte d'une réponse d'incident. Pour la collecte et l'analyse il existe deux scénarios possibles. Soit, la machine est toujours en état de fonctionnement et il sera alors possible d'effectuer une capture de la mémoire vive ou d'utiliser une série de commandes afin de lister les processus et services en cours, soit la machine est éteinte et il faudra alors consulter les journaux et le registre afin de tenter de reconstituer l'état de la machine à posteriori. Le premier scénario est évidemment idéal, mais il n'est pas toujours possible de garder une machine en fonctionnement¹².

Afin de déterminer si des processus malveillants ou inhabituels opèrent sur un système, il faut au préalable connaître l'état normal dudit système. Nous allons donc nous

12. Si l'on suspecte par exemple qu'elle est infectée par un logiciel malveillant et que celui-ci risque de se propager ou tout simplement si la machine a déjà été éteinte par un utilisateur ou par l'attaquant.

pencher sur l'arbre de création des processus au démarrage de Windows et aborder rapidement le rôle de chaque composant. La figure 3.2 offre une représentation visuelle de cet arbre.

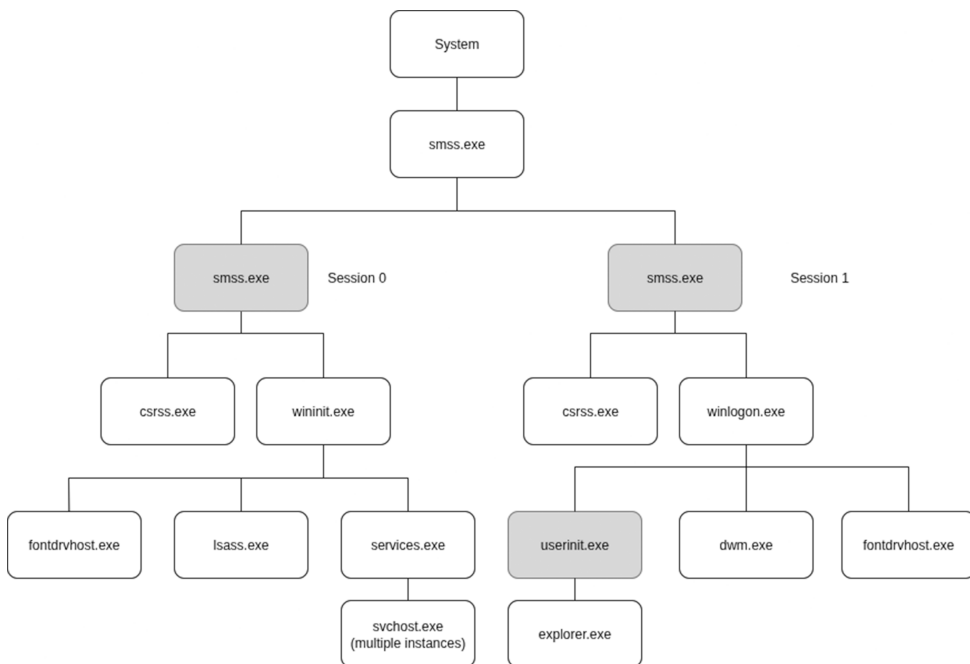


FIGURE 3.2 – Arbre des processus par défaut d'un système Windows.

Cet arbre est volontairement tronqué pour le rendre plus lisible, mais les principaux processus y sont représentés. Les processus en blanc représentent ceux qui restent actifs une fois le processus de démarrage terminé alors que ceux en gris illustrent ceux qui se terminent une fois leur tâche accomplie. Comme racine de notre arbre, nous retrouvons le processus "System" qui est en charge des principaux *threads* liés au noyau du système comme le chargement des différents pilotes et des modules de base de Windows. Un des enfants de ce processus est "smss.exe", la composante responsable de la gestion des sessions. Une première session parente est créée et sera responsable du démarrage d'une instance pour chaque nouvelle session. Il y aura toujours au minimum une session 0 réservée au fonctionnement du système. Ensuite, une nouvelle session sera créée pour chaque utilisateur connecté à la machine. Le gestionnaire de session va ensuite démarrer d'autres processus (*wininit.exe* pour la session 0 ou *winlogon* pour une session utilisateur) avant de se terminer. Pour la session système, le processus *wininit.exe* va initier plusieurs éléments importants de Windows comme *service.exe*, le gestionnaire des processus qui seront exécutés en tâche de fond ainsi que *lsass.exe*, le processus responsable de l'authentification des utilisateurs. Le processus *services.exe* sera en charge de lancer *svchost.exe*, un processus générique qui sera utilisé pour charger des bibliothèques critiques sous Windows (fichier .dll). On y retrouve notamment *taskhostw.exe*, le processus responsable des tâches planifiées. Notons qu'on observera un nombre important d'instances de *svchost.exe* sur une machine (entre 20 et 50). Cela résulte du fait que pour

chaque librairie utilisée, ce processus parent sera utilisé. Nous verrons plus loin comment malgré tout différencier les instances légitimes de celles qui sortent de l'ordinaire. Du côté utilisateur, le processus *winlogon.exe* permet à un utilisateur de s'authentifier et d'exécuter *explorer.exe* au travers d'un processus temporaire *userinit.exe*. *explorer.exe* correspond à l'explorateur Windows, c'est-à-dire à l'ensemble des fonctionnalités qui permettent à un utilisateur d'interagir avec le système de fichiers. On retrouve également des exécutables qui sont lancés dans toutes les sessions comme *csrss.exe* et *fontdrvhost.exe*. Le premier s'occupe de gérer des processus et des *thread*, notamment en chargeant un nombre important de *dll* liées à l'API Windows. Le deuxième s'occupe de gérer les polices sur le système.

Le tableau 3.4 liste et résume le rôle des principaux processus sollicités pour un fonctionnement classique de Windows (liste non exhaustive).

Processus	Rôle principal
System	Gestion des <i>threads</i> du noyau, chargement des pilotes et modules de base
smss.exe	Gestion des sessions Windows et initialisation des processus système
wininit.exe	Initialisation des composants système critiques
services.exe	Gestion et démarrage des services Windows en arrière-plan
lsass.exe	Authentification des utilisateurs et gestion de la sécurité
svchost.exe	Processus parent pour la plupart des processus liés aux services
winlogon.exe	Gestion de l'authentification utilisateur et du démarrage de session
userinit.exe	Initialisation de l'environnement utilisateur et lancement d' <i>explorer.exe</i>
explorer.exe	Interface utilisateur et interaction avec le système de fichiers
csrss.exe	Gestion des processus, des <i>threads</i> et de la console
fontdrvhost.exe	Gestion des polices du système

TABLE 3.4 – Principaux processus Windows et leur rôle.

Afin de visualiser les informations relatives à tous les processus en cours sur une machine, nous commencerons par utiliser l'outil *System Informer*¹³. Celui-ci permet d'afficher un grand nombre de paramètres pour chaque processus et offre une vue en arbre très utile pour comprendre la structure et les interactions entre ceux-ci. La figure 3.3 montre par exemple une illustration de l'arborescence des différents processus discutés jusqu'ici.

À gauche de l'écran, on retrouve le nom des processus ainsi qu'une arborescence qui permet de mettre en évidence les processus parents. La colonne au centre donne, quant à elle, le nom de l'utilisateur qui a initié le processus. Pour la plupart, il s'agit du système lui-même. En bas de la capture, on aperçoit une série de processus lancés par l'utilisateur fictif "*victim*". On y retrouve le démarrage de l'explorateur de fichiers, ainsi qu'une invite de commande "*cmd.exe*", un processus lié à l'outil "*onedrive*" et pour finir le processus lié à notre outil *System Informer*. À droite de l'écran, il est possible d'afficher la ligne de commande complète qui a permis de lancer chaque processus. Cette information est précieuse, car cela permet de lier le nom d'un processus avec le binaire qui est réellement

13. <https://systeminformer.sourceforge.io/>

System Informer [DESKTOP-LF6HGS\victim] (Administrator)						
System View Tools Users Help						
Refresh Options Find handles or DLLs System information						
Processes	Services	Network	Disk	Firewall	Devices	
Name	PID	2.5... CPU	92 B/s I/O total r...	1,99 GB Private b...	User name	Description
System Idle Process		97,46		60 kB	NT AUTHORITY\SYSTEM	
System	4	0,46		40 kB	NT AUTHORITY\SYSTEM	NT Kernel & System
Registry	112			6,42 MB	NT AUTHORITY\SYSTEM	
smss.exe	456			1,12 MB	NT AUTHORITY\SYSTEM	Windows Session Manager
Memory Compression	1688			608 kB	NT AUTHORITY\SYSTEM	\\SystemRoot\System32\smss.exe
Interrupts		0,92		0		Interrupts and DPCs
csrss.exe	624			1,97 MB	NT AUTHORITY\SYSTEM	Client Server Runtime Process
wininit.exe	696			1,48 MB	NT AUTHORITY\SYSTEM	Windows Start-Up Application
csrss.exe	716	0,03		2,04 MB	NT AUTHORITY\SYSTEM	Client Server Runtime Process
winlogon.exe	768			2,48 MB	NT AUTHORITY\SYSTEM	Windows Logon Application
fontdrvhost.exe	1012			1,88 MB	Font Driver Host\UMFD-1	Usermode Font Driver Host
dwvm.exe	1092	0,16		70,51 MB	Window Manager\DWVM-1	Desktop Window Manager
explorer.exe	5016	0,03		38,16 MB	DESKTOP-LF6HGS\victim	Windows Explorer
cmd.exe	11228			2,57 MB	DESKTOP-LF6HGS\victim	Windows Command Processor
MoNotificationUx.exe	7956			2,26 MB	DESKTOP-LF6HGS\victim	Notifications host process
OneDrive.Sync.Service.exe	2932			37,67 MB	DESKTOP-LF6HGS\victim	Microsoft OneDrive Sync Service
SystemInformer.exe	11112	0,44		25,68 MB	DESKTOP-LF6HGS\victim	System Informer

exécuté. En déployant l'onglet du processus *explorer.exe*, on aperçoit une liste de sous-processus directement démarrés à l'aide de l'interface utilisateur, comme illustré sur la figure 3.4.

Process	PPID	Parent Process	Architecture	Path	Command Line
explorer.exe	5016	Desktop-LF6H85\victim	x64	C:\WINDOWS\Explorer.exe	
SecurityHealthSy...	8376	Desktop-LF6H85\victim	x64	C:\Windows\System32\SecurityHealth\Systray.exe	
VBoxTray.exe	8452	84 B/s	x64	C:\Windows\System32\VBoxTray.exe	
OneDrive.exe	8608	Desktop-LF6H85\victim	x64	C:\Users\victim\AppData\Local\Microsoft\OneDrive\OneDrive.exe	/background
ms-teams.exe	9796	Desktop-LF6H85\victim	x64	C:\Users\victim\AppData\Local\Microsoft\WindowsApps\MSTeams_8595b643bb7bwlms-teams.exe	msteams-system-initiated
msedgeviewb...	9992	Desktop-LF6H85\victim	x64	C:\Program Files (x86)\Microsoft\EdgeWebView\Application\142.0.3595.65\msedgeviewbview2.exe	--embedded-browser-viewbiew1-1-we...
msedge.exe	10016	Desktop-LF6H85\victim	x64	C:\Program Files (x86)\Microsoft\EdgeWebView\Application\142.0.3595.65\msedgeviewbview2.exe	--types:crashpad-handler --type-data-d...
msedge.exe	10176	Desktop-LF6H85\victim	x64	C:\Program Files (x86)\Microsoft\EdgeWebView\Application\142.0.3595.65\msedgeviewbview2.exe	--types:gpu-process --gpu-watcher-d...
msedge.exe	10188	Desktop-LF6H85\victim	x64	C:\Program Files (x86)\Microsoft\EdgeWebView\Application\142.0.3595.65\msedgeviewbview2.exe	--type:utility --utility-sub-types:netwo...
msedge.exe	8004	Desktop-LF6H85\victim	x64	C:\Program Files (x86)\Microsoft\EdgeWebView\Application\142.0.3595.65\msedgeviewbview2.exe	--type:utility --utility-sub-types:stora...
msedge.exe	8772	Desktop-LF6H85\victim	x64	C:\Program Files (x86)\Microsoft\EdgeWebView\Application\142.0.3595.65\msedgeviewbview2.exe	--types:renderer --noerialdialogs --user...
SystemInform...	9204	Desktop-LF6H85\victim	x64	C:\Program Files\SystemInform\er\SystemInform.exe	
cmd.exe	11228	Desktop-LF6H85\victim	x64	C:\WINDOWS\system32\cmd.exe	&&C:\Users\Public\Documents\node\startnode.bat"
conhost.exe	12136	Desktop-LF6H85\victim	x64	C:\WINDOWS\system32\conhost.exe	/V/C:\Users\Public\Documents\node\startnode.bat
conhost.exe	7800	Desktop-LF6H85\victim	x64	C:\Users\Public\Documents\node\node.exe	&&C:\Users\Public\Documents\node\startnode.bat
MoNotificationUx...	7952	Desktop-LF6H85\victim	x64	C:\WINDOWS\ui\packages\groveui\AMD64\MoNotificationUx\NotificationType Scan_Seeker_Available/FormFactor Passive/CV pM...	
OneDrive.Sync.Serv...	2936	Desktop-LF6H85\victim	x64	C:\Users\victim\AppData\Local\Microsoft\OneDrive\25.199.1012.001\OneDrive.Sync.Service.exe	/silent/Config
msedge.exe	2404	224 B/s	x64	C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	--no-startup-window
msedge.exe	3576	Desktop-LF6H85\victim	x64	C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	--types:crashpad-handler --type-data-dir=C:\Users\victim\AppData\Lo...
msedge.exe	10948	Desktop-LF6H85\victim	x64	C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	--types:gpu-process --no-pre-read-main-di... --gpu-preferences:SAAAA...
msedge.exe	11084	224 B/s	x64	C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	--type:utility --utility-sub-types:network.mojom.NetworkService --lang...

Notez que certaines applications ne semblent pas être liées au processus *explorer.exe*. En réalité, la source de l'exécution d'un programme va déterminer son emplacement dans l'arborescence des processus. *System Informer* a par exemple été lancé depuis le bureau alors que l'application *Node.js* est lancé par *cmd.exe* lorsque l'utilisateur se connecte. Il ne faut donc pas limiter les recherches aux processus enfants de l'explorateur de fichiers. En effet, même si certains processus sont liés à un utilisateur, cela ne signifie pas pour autant que c'est celui-ci qui les a lui-même lancés. En réalité, beaucoup d'entre eux sont automatiquement démarrés par le système, mais "pour le compte" de l'utilisateur spécifié dans la capture d'écran. Le processus héritera alors de permissions de celui-ci.

14. Attention, l'outil *System Informer* doit être lancé en mode administrateur pour accéder à toutes ces informations

wininit.exe	696	wininit.exe
services.exe	844	C:\WINDOWS\system32\services.exe
svchost.exe	984	C:\WINDOWS\system32\svchost.exe -k DcomLaunch -p
svchost.exe	628	C:\WINDOWS\system32\svchost.exe -k RPCSS -p
svchost.exe	816	C:\WINDOWS\system32\svchost.exe -k DcomLaunch -p -s LSM
svchost.exe	1132	C:\WINDOWS\System32\svchost.exe -k LocalServiceNetworkRestricted -p -s lmhosts
svchost.exe	1272	C:\WINDOWS\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s TimeBrokerSvc
svchost.exe	1280	C:\WINDOWS\System32\svchost.exe -k LocalSystemNetworkRestricted -p -s NcbService
svchost.exe	1308	C:\WINDOWS\system32\svchost.exe -k netsvcs -p -s Schedule
svchost.exe	1360	C:\WINDOWS\system32\svchost.exe -k LocalServiceNoNetwork -p
svchost.exe	1392	C:\WINDOWS\system32\svchost.exe -k UserProfileService -p -s ProfSvc
svchost.exe	1404	C:\WINDOWS\system32\svchost.exe -k LocalService -p -s nsi
svchost.exe	1500	C:\WINDOWS\system32\svchost.exe -k LocalSystemNetworkRestricted -p -s DevQueryBroker
svchost.exe	1520	C:\WINDOWS\System32\svchost.exe -k netprofm -p -s netprofm
svchost.exe	1556	C:\WINDOWS\system32\svchost.exe -k netsvcs -p -s UserManager
svchost.exe	1572	C:\WINDOWS\system32\svchost.exe -k LocalService -p -s DispBrokerDesktopSvc
svchost.exe	1832	C:\WINDOWS\system32\svchost.exe -k NetworkService -p

FIGURE 3.5 – Arguments passés à l'exécution d'une instance *svchost.exe*.

On y retrouve pour chaque instance de *svchost.exe* le chemin vers l'exécutable ainsi que les paramètres passés à l'exécution. Ces paramètres s'appuient sur la clé de registre suivante pour opérer :

- 1 Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
 → NT\CurrentVersion\Svchost

Cette clé de registre structure l'ensemble des services en différents groupes. Historiquement, pour des raisons de performances, une instance de *svchost.exe* était partagée pour l'exécution de plusieurs services membres d'un groupe défini dans la clé de registre *Svchost*. Depuis la version *Windows 10 1703*, Microsoft a décidé d'établir une instance *svchost* par service¹⁵, afin d'améliorer la sécurité et la fiabilité du fonctionnement de ses services¹⁶. Pour lier un processus *svchost.exe* à une application qui tournera en tâche de fond, on utilise donc le paramètre *k* pour déterminer le groupe de services et le paramètre *s* pour spécifier le service.¹⁷. Par exemple, en consultant la clé de registre *Svchost*, on y retrouve le groupe *UnistackSvcGroup* et dans cette liste, le service *CDPUserSvc* (voir figure 3.6).

La commande liée à l'instance *svchost* sera donc la suivante :

- 1 C:\WINDOWS\system32\svchost.exe -k UnistackSvcGroup -s CDPUserSvc

Nous pouvons maintenant lier une instance générique *svchost* à un nom de service, mais nous n'avons toujours pas l'information qui spécifie le chemin de l'exécutable ou de la librairie qui sera exécuté. Il faut pour cela utiliser l'identifiant du processus (*process id*, ou *PID*) pour faire le lien entre une instance *svchost* et le service. Dans notre exemple précédent, le service *CDPUserSvc* est identifié avec le *PID* 1092. Dans l'onglet *Services* de *System Informer*, on retrouve facilement le service concerné, soit à l'aide du *PID* où en s'aidant du nom de celui-ci. Il est ensuite possible de récupérer des détails plus précis concernant la librairie utilisée comme illustré sur la figure 3.7.

15. Si la quantité de mémoire vive est suffisante, c'est-à-dire supérieure à 3,5Gb

16. <https://learn.microsoft.com/en-us/windows/application-management/svchost-service-refactoring>

17. L'argument *-p* semble lui définir une politique spécifique lors de l'exécution du service, mais Microsoft Windows restant un système propriétaire, il n'existe pas de documentation officielle sur le code source de la partie système. Microsoft propose par contre une documentation de son API, utile pour les développeurs d'applications pour ce système d'exploitation

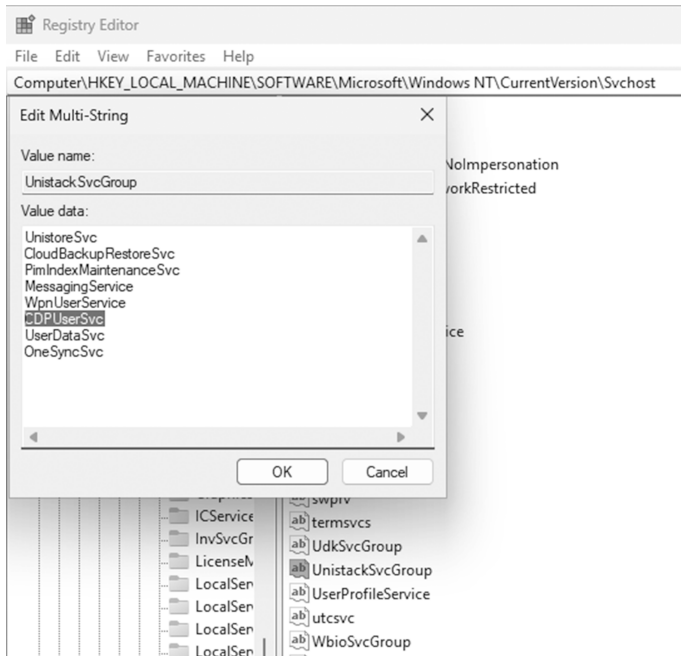


FIGURE 3.6 – Gestion des groupes de services dans le registre Svchost.

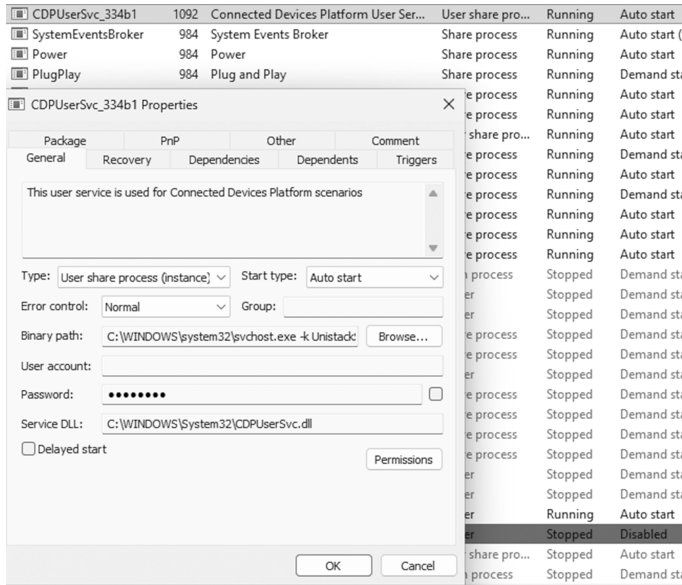


FIGURE 3.7 – Détails d'un service dans *System Informer*.

Dans les détails d'un service, est indiqué l'emplacement, à la fois du binaire de *svchost*, mais également celui de la librairie qui sera chargée, ici : *CDPUserSvc.dll*. Malheureusement toutes les librairies systèmes de Windows ne sont pas forcément bien documentées. Par contre, il est possible de vérifier si celle-ci a bien été signée par l'autorité

de certification de Microsoft. Pour cela, il suffit d'effectuer un clic-droit sur le fichier *.dll* et de consulter le certificat. La chaîne d'autorité permet d'assurer la validité du fichier comme illustré sur la figure 3.8.

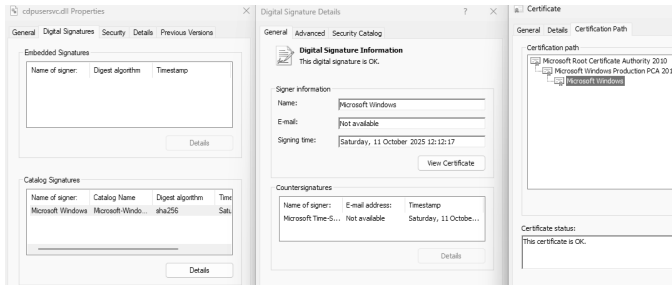


FIGURE 3.8 – Vérification du certificat d'une librairie système.

Il existe également une application en ligne de commande développée par Microsoft, *sigcheck*¹⁸ qui permet de vérifier rapidement la validité du certificat d'un exécutable ou d'une librairie. Ce programme permet également d'effectuer des requêtes vers *VirusTotal* afin de s'aider de son système de détection.

Voler des certificats pour signer des logiciels malveillants - le cas de Stuxnet

Un système de défense comme *Windows Defender* ne peut pas bloquer l'exécution de chaque nouvelle application sur une machine. Parmi les différentes techniques de détection utilisées, *Windows Defender* va vérifier le certificat qui a permis de signer un exécutable afin de s'assurer qu'il puisse faire confiance à celui-ci. S'il s'agit d'une autorité de certification reconnue, alors l'application scannée sera considérée comme légitime. Toutefois, certains groupes d'attaquants ont parfois réussi à voler des certificats à des entreprises que Windows considère comme de confiance afin de signer des programmes malveillants. C'est par exemple ce qu'ont effectué les concepteurs de *Stuxnet* en 2010 en volant des certificats à l'entreprise *Realtek* pour rendre leur *malware kernel* indétectable par les systèmes de défense des victimes^a. Même s'il s'agit d'attaques avancées et rares, il faut garder à l'esprit qu'un logiciel signé n'est pas une garantie absolue pour conclure qu'il s'agit d'un processus légitime.

^a. <https://en.wikipedia.org/wiki/Stuxnet>

En résumé, la difficulté dans l'analyse des processus en cours sur une machine (que ce soit sur un système en fonctionnement ou en travaillant avec une capture de mémoire vive) réside dans la distinction de ce qu'est un processus légitime et ce qui ne l'est pas. Premièrement, il existe un nombre important de processus et services utilisés par le système en lui-même. Il est donc important de pouvoir les identifier rapidement afin de les exclure d'une analyse. De plus, des applications malveillantes peuvent prendre le nom de services ou processus connus afin de tromper l'analyste. Il est donc parfois utile de

18. <https://learn.microsoft.com/en-us/sysinternals/downloads/sigcheck>

constituer une "*white list*" de noms de processus à partir d'un système sain. Comparer les résultats permettra de mettre en évidence un delta potentiellement malveillant. Notons également que des *malwares* peuvent utiliser différentes techniques afin de remplacer des exécutables ou des bibliothèques existantes en conservant le même nom. C'est pour cette raison qu'il est important de vérifier l'empreinte et les certificats des fichiers binaires à la source de processus suspects. Une autre technique utilisée par les *malwares* consiste à effectuer une injection dans la partie de la mémoire réservée à un processus légitime afin d'y exécuter du code malveillant. Ce type de technique nécessitera une analyse plus poussée en inspectant les espaces mémoires utilisés par des processus avec des outils comme *Volatility*. Dans le chapitre dédié à l'analyse d'un cas réel, nous développerons ces techniques afin de montrer comment identifier des processus malveillants dans une capture de la mémoire vive.

4 Activité réseau et connexions à distance

Un ordinateur est rarement utilisé de manière isolé. La plupart du temps chaque poste utilisateur sera connecté au réseau interne de l'entreprise et pourra également communiquer avec l'extérieur via Internet. De plus, une série de services internes à Windows utilisent les protocoles réseau pour des applications qui communiquent localement au sein d'une même machine. Pour finir, il est également possible de se connecter à distance sur une machine, afin de faciliter l'administration ou l'organisation du travail. L'échange d'informations par le réseau est l'un des principaux enjeux de la sécurité informatique pour ne pas dire le plus grand. En effet, il est important que ces communications restent privées tout en garantissant leur intégrité et l'authentification des émetteurs et récepteurs de l'information. De plus, c'est le plus souvent à travers le réseau que les attaquants vont pouvoir entrer dans l'infrastructure d'une victime et s'y propager. Lors d'une réponse à un incident, il sera nécessaire d'identifier les différentes connexions existantes et passées dans une infrastructure afin d'y détecter de possibles anomalies. Notons qu'ici nous nous intéresserons exclusivement aux connexions identifiables sur une machine Windows. L'analyse du trafic à partir d'une capture sera discutée dans la partie 3 de ce livre.

TCP et UDP

Les protocoles de transport *TCP* et *UDP* permettent d'encapsuler des données en y ajoutant une adresse de destination (adresse IP, qui permet de différencier les machines) ainsi qu'un numéro de port (permettant de différencier les applications qui tournent sur une machine). Le protocole *TCP* établit une connexion préalable entre les deux machines qui veulent échanger des paquets afin d'assurer la retransmission des paquets qui seraient perdus ou corrompus durant l'échange. *UDP* lui n'utilise pas de système de connexion. Il est généralement considéré comme plus rapide, mais moins fiable.

Identifier le trafic légitime

Comme pour les autres indicateurs de compromission, la tâche la plus difficile pour un analyste consistera à différencier les éléments qui sont propres au fonctionnement d'un système Windows ou des applications légitimes installées sur celui-ci, avec ceux

LA RÉPONSE D'INCIDENT

Analyser et réagir face à une cyberattaque

Sécuriser son infrastructure face aux cyberattaques n'est plus suffisant. Il faut se préparer au pire.

À la suite d'une cyberattaque, les entreprises ou les institutions publiques doivent se redresser. Pour cela, il est avant tout nécessaire de comprendre ce qui s'est passé.

- Qui les a attaquées et pourquoi ?
- Quelles failles techniques ou humaines ont été exploitées pour prendre le contrôle de leur infrastructure ?
- Quelles données ont été exfiltrées ?

L'objectif de ce livre est de montrer comment il est possible, à partir d'une masse d'informations, de trouver les éléments importants qui permettront de comprendre une attaque et de potentiellement récupérer des données compromises. Pour rencontrer ces objectifs, l'ouvrage s'articule autour de trois parties qui répondent respectivement aux questions : « Quelles preuves récolter ? Comment les récolter ? Comment les analyser ? »

Ce livre propose une approche innovante dans laquelle le lecteur suivra une enquête à travers un scénario réaliste de cyberattaque. Le lecteur sera également invité à mener lui-même cette analyse grâce aux différents exercices et ressources mis à disposition par les auteurs. De plus, ce livre est accompagné de recommandations de lectures supplémentaires afin d'approfondir certaines thématiques.

Adrien Voisin : Diplômé en sciences informatiques à l'Université Catholique de Louvain, il est Maître-Assistant à La Haute École de Namur-Liège-Luxembourg (Henallux) où il y enseigne la sécurité offensive et défensive. Il a participé à la création de l'un des premiers cours de réponse aux incidents et d'analyse forensique pour l'enseignement supérieur en Belgique.

Guillaume Duvillé : Docteur en informatique de l'Université de Montpellier, il a enseigné l'investigation numérique et la sécurité des systèmes à la Haute-École de Namur Liège-Luxembourg (Henallux) avant d'occuper le poste d'architecte cybersécurité pour le compte du Service Public de Wallonie.

21,90 €

ISBN : 978-2-8073-6979-5



9 782807 369795

De Boeck Supérieur s'engage
durablement
pour l'environnement



DANS LA MÊME
COLLECTION



www.deboecksuperieur.com