L'investigation numérique « libre »

Sébastien LARINIER[1], Solal JACOB[2]

[1]Atheos, 75 avenue Victor HUGO, 92500 Rueil Malmaison

[2]ArxSys, 14-16 rue Soleillet, 75020 Paris

slarinier@atheos.fr, sja@arxsys.fr

Résumé – Cet article a pour but de présenter rapidement l'investigation numérique classique et à source ouverte, les méthodologies à appliquer et les avantages des logiciels libres dans ces domaines.

Abstract – This article gives a brief overview of what are digital forensics and OSINT (Open Source INTelligence), which methodology should be applied, and the benefits of using Open Source software in those fields.

1. L'investigation numérique

Les techniques d'investigation n'ont cessé d'évoluer depuis la première apparition de la médecine légale en 1247, dans « Xi Yuan Ji Lu » 洗冤集錄, *Recueil pour laver les injustices,* de Song Ci 宋慈 [1].

Aujourd'hui, l'utilisation massive des technologies de l'information a conduit à la création d'une nouvelle branche dans ce domaine : l'investigation numérique aussi appelée informatique légale (computer forensic en anglo-saxon).

1.1 La criminalité du XXI^{e} siècle

Les outils de communication actuels ont donné naissance à un nouveau type de criminalité, le cybercrime, qui se décline sous plusieurs formes.

1.1.1 Les différents types de cybercrime.

On trouve :

- Tout d'abord des infractions ***spécifiques*** aux technologies de l'information et de la communication. Ceci correspond aux atteintes aux systèmes de traitements automatisés de données, aux données à caractère personnel et aux interceptions.
- Puis les infractions ***liées*** aux technologies de l'information et de la communication, telles que la pédopornographie, l'incitation à la haine raciale sur Internet ou les atteintes aux personnes et aux biens.
- Enfin, les infractions ***facilitées*** par les technologies de l'information et de la communication telles que l'escroquerie en ligne, la contrefaçon ou toute autre violation de la propriété intellectuelle.

1.2 Des méthodologies légiféré

Pour répondre aux besoins des enquêteurs spécialisés dans les domaines technologiques et permettre de valider les preuves recueillies, des méthodologies ont été développées au fil du temps.

Leur but est de décrire les étapes nécessaires pour assurer l'authenticité, la fiabilité et l'intégrité des preuves, qui sont des critères exigés par les magistrats.

1.2.1 Les étapes clés

La méthodologie la plus utilisée est dénommée « Investigative Process Model » [2]. Elle permet de couvrir toute investigation et non pas uniquement la partie numérique.

Les étapes principales étant la préservation, l'acquisition et/ou la récupération de données, la réduction, la recherche, l'analyse et enfin la création d'un rapport.

Ces différentes étapes doivent être effectuées avec minutie et peuvent prendre énormément de temps, or les enquêteurs, lors des perquisitions, ou de gardes à vue, n'en disposent parfois que de très peu.

Pour s'assurer de la validité de ces étapes et accélérer leur traitement, de nombreux outils ont vu le jour dans le but d'assister les enquêteurs.

2. Les outils

Différents types d'outils existent et chacun s'adresse à une partie spécifique de la méthodologie précédemment décrite. Par exemple : des outils matériels appelés couramment « bloqueurs en écriture » permettent, comme leur nom l'indique, de prévenir toute écriture lors de la phase d'acquisition d'un support numérique tel qu'un disque dur, cela afin de produire un clone du support d'origine et de pouvoir travailler dessus.

Or, les outils ayant le plus évolué ces dernières années sont les outils logiciels.

2.1 Une évolution perpétuelle

Au début des années 1980, aucun outil n'était réellement dédié à l'investigation numérique, les premiers enquêteurs utilisent alors des programmes grands publics pour accéder au contenu des supports numériques. Un des outils les plus populaires alors, était Norton Utilities (1982) qui fournissait un programme nommé « undelete » qui permettait de retrouver des fichiers supprimés sur les systèmes de fichiers FAT.

2.1.1 Les outils dédiés

Les premiers outils dédiés furent créés en 1992 par la société américaine ASR Data (Texas) puis d'autres suivirent. À l'époque, les outils étaient épars, disposant chacun d'une fonctionnalité dédiée. Il fallut attendre encore quelques années pour qu'ASR créée le premier logiciel regroupant un maximum de fonctionnalités au sein d'une interface unique : « ASR Expert Witness » était née et fut rachetée en 1997 par Guidance Software et devint Encase, toujours leader du marché actuel.

Quelques années plus tard vinrent les premiers concurrents tels que « Forensic Tool Kit » d'Access Data (USA) et l'allemand X-Ways Forensics basé sur le populaire « WinHex », un éditeur hexadécimal permettant d'analyser un fichier octet par octet.

2.2 De l'importance de l'Open Source

Les logiciels propriétaires précédemment cités sont encore majoritairement utilisés lors d'investigations numériques. Pourtant depuis 1999, une alternative est apparue sous forme de logiciel Open Source. Le premier fut « TCT » (The Coroner's Toolkit) de Dan Farmer & Wietse Venema, aussi auteurs d'un des premiers livres sur le sujet [3]. Or, comme nous allons le voir, les logiciels Open Source offrent une véritable alternative et apportent une plus-value face à leurs homologues propriétaires.

2.2.1 Daubert contre Merell Dow Pharmaceuticals

En 1993, le procès de Merell Dow Pharmaceuticals conte Daubert [4] fut une véritable révolution aux USA, car suite à ce procès une nouvelle jurisprudence « Daubert Standard » vint remplacer le « Frye standard » [5].

Cette jurisprudence établit que la partie qui présente une preuve fondée sur une analyse scientifique ou sur l'utilisation d'une technique doit, à priori, en établir la validité, sous peine de la voir exclue, en répondant au moins à ses 4 points précis :

- Est-ce que la théorie sous-jacente à ladite science ou la technique a été mise à l'essai ou est-elle falsifiable ?
- Quel est le taux d'erreur établi ou potentiel ?
- Est-ce que la théorie ou la technique a été évaluée par des pairs et a été sujette à publication ?
- Existe-t-il un consensus au niveau de la communauté scientifique quant à la fiabilité de ladite technique et de la théorie fondamentale ?

Or, s'il est possible de répondre à toutes ces questions lors de l'utilisation d'un logiciel à source ouverte, il en est impossible lors de l'utilisation d'un logiciel fermé [6], dont le code source n'est pas librement diffusé.

Les logiciels fermés agissent en effet comme une boîte noire et les sociétés qui les développent font tout pour cacher leur fonctionnement interne. Par exemple, les logiciels Encase et FTK utilisent une technique de chiffrement de code [7] non cassé par les experts en rétro-ingénierie à l'heure actuelle. Il est impossible de savoir de quoi sont composés ces logiciels, comment ils traitent et analysent l'information et s'ils ne la déforment pas. C'est pourtant un point essentiel pour s'assurer de la validité d'une preuve.

2.2.2 Le code libre donne-t-il un avantage aux cybercriminels ?

Un argument souvent avancé face aux logiciels libres dans le monde de l'investigation numérique est que les attaquants ayant accès au code source des logiciels peuvent en détourner plus facilement les fonctionnalités.

Or, sans prendre en compte que les logiciels propriétaires utilisent souvent eux-mêmes du code libre [8], en 2007, lors de la conférence sur la sécurité informatique Black Hat, ISEC Partners présenta un article nommé « Breaking Forensics Software: Weaknesses in Critical Evidence Collection » [9].

Dans cet article des outils Open Source et propriétaires sont étudiés de manière similaire, le but étant de trouver différentes failles dans ces logiciels. Pour ce faire, différentes techniques ont été utilisées telles que : l'injection de données aléatoires, la création de systèmes de

fichiers contenant des boucles sur les répertoires, ou la création de très nombreuses partitions.

Les résultats concluent que les deux types de logiciels contiennent des failles. Les logiciels d'investigation disposant de nombreuses fonctionnalités et étant composés d'énormément de code, ceci n'est au final, que peu étonnant. Mais certaines de ces failles, en plus de faire « planter » les logiciels, permettent de cacher des données à l'enquêteur ce qui pose évidemment un réel problème.

La conclusion de l'article est que la réponse et la correction des failles a été étonnement bien plus rapide sur les logiciels Open Source testés que sur les logiciels propriétaires. Le développement communautaire et le mode de distribution Open Source permettant sûrement une meilleure réactivité quant au mode de développement propriétaire qui est moins réactif et nécessite une mise en place de suivi et de « patch » bien plus fastidieuse.

Ceci démontre donc que malgré les idées reçues les logiciels Open Source ne donnent en aucun cas un avantage au cybercriminel en publiant leur code, et au contraire que les logiciels propriétaires peuvent eux aussi être manipulés, voire déjoués.

3. Digital Forensics Framework

Comme nous l'avons vu précédemment, malgré les avantages indéniables des solutions Open Source face à leurs équivalents propriétaires, ces derniers sont toujours majoritairement utilisés par les enquêteurs. Ils ne peuvent alors répondre au besoin de transparence nécessaire à une investigation numérique.

3.1 Vers une solution Open Source viable

Lorsque l'on étudie le marché des logiciels d'investigation, il s'avère que la majeure partie des logiciels Open Source existants ne sont pas soutenus par des entreprises. Il est donc difficile pour les enquêteurs d'obtenir une hotline, les formations et les certifications requises pour de tels produits.

De plus, la plupart de ces logiciels ne sont disponibles que sous un environnement Linux ou UNIX, alors que les enquêteurs travaillent encore majoritairement sur les environnements Windows®.

3.1.1 La genèse

En 2007, un groupe d'étudiants de l'EPITECH fait le choix de concevoir un logiciel libre d'investigation numérique dans le cadre d'un projet de fin d'études en partenariat avec l'IRCGN (Institut de Recherche Criminelle de la Gendarmerie Nationale).

Face à la demande qui commence à se faire ressentir d'une professionnalisation des logiciels Open Source d'investigation, le groupe d'étudiants à l'origine de DFF décide de créer une entreprise, ArxSys, dans le but de continuer le développement du logiciel et de proposer des services attendus par les enquêteurs.

3.2 Un cadre de travail modulaire et évolutif

Le but de DFF est de fournir un cadre de travail regroupant la majorité des fonctionnalités nécessaires lors d'une enquête sur des éléments de preuves numériques, tout en préservant la validité de ces preuves afin d'en assurer la validité face à une cour de justice.

Pour cela, le logiciel se divise en trois parties majeures.

3.2.1 Une interface de programmation unifiée

DFF dispose d'une API (interface de programmation) dans le but d'unifier le développement du logiciel. Ceci est un point clé pour un logiciel Open Source car cette API permet aux développeurs de se décharger de nombreuses tâches et de ne pas développer plusieurs fois le même code.

L'API de DFF est elle-même composée de nombreuses bibliothèques de programmation dont un système de fichiers virtuels pour représenter l'arborescence des fichiers, d'une bibliothèque d'identification de type de fichier, d'un moteur de recherche appliqué aux noms et contenus des fichiers, ou encore d'un gestionnaire de tâches et de modules.

3.2.2 Les modules

Les modules ou « greffons » permettent de développer des fonctionnalités dédiées de manière simplifiée en se basant sur l'API. Ceux-ci sont indépendants et peuvent être développés sans une connaissance totale du logiciel. Leur but est d'encourager la contribution, mais ils facilitent aussi la distribution entre enquêteurs de scripts leurs permettant d'extraire des informations essentielles lors de leurs enquêtes, et ainsi de répondre à un besoin spécifique dans un cas précis.

Par défaut, DFF fournit de nombreux modules (une soixantaine) pour analyser des systèmes de fichiers (FAT, NTFS, EXTFS), pour extraire des métadonnées de documents (MSDOC, MSPPT, EXIF), pour reconstruire des machines virtuelles ou encore pour vérifier l'intégrité des données (hash MD5, SHA1…).

3.2.3 Les interfaces utilisateurs

Les enquêteurs numériques possèdent en général une connaissance technique pointue aussi bien dans le domaine informatique que juridique, mais la plupart ne possèdent pas de connaissances en développement. Ils ont aussi besoin d'une interface simple et rapide d'utilisation leur permettant de visualiser le contenu des documents.

Pour cela DFF propose trois interfaces, une interface graphique étudiée pour être simple d'utilisation, une interface en ligne de commande pour être utilisée sur un serveur et enfin un interpréteur dédié aux développeurs et utilisateurs chevronnés permettant de scripter le logiciel en direct.

3.3 Communauté, recherche et évolutions

Un projet Open Source comme DFF, en plus d'être soutenu par une entreprise pour assurer un service fiable, a besoin d'une communauté pour se faire connaître, se développer et s'améliorer.

3.3.1 Service communautaire

Le code source de DFF est disponible par le biais d'un logiciel de gestion de versions décentralisé [10] qui permet le travail collaboratif sur le logiciel.

Pour faciliter la communication entre les contributeurs, trois listes de diffusion ont été mises en place [11] ainsi qu'un gestionnaire de projets [12].

3.3.2 Recherche et développement

Les logiciels Open Source sont particulièrement adaptés au monde de l'enseignement et de la recherche, d'une part du fait de leur gratuité, et d'autre part parce qu'ils permettent aux chercheurs de développer et d'intégrer leur preuve de concept.

DFF a entre autres permis la résolution d'un concours créé lors de la célèbre conférence DFRWS (Digital Forensics Research Workshop) [13] et la réalisation d'une thèse par un étudiant de l'université de Mannheim (Allemagne).

4. Transition

On vient de voir un comparatif entre un logiciel commercial et un logiciel Open Source. A présent, nous allons mettre en évidence les limites de certains logiciels et voir comment il est possible de combler certains manques.

Le but d'un enquêteur numérique est de réunir le plus d'informations possible pour comprendre ce qu'il est en train d'observer et déterminer le mode opératoire potentiel ainsi que les motivations d'un attaquant lorsqu'il s'agit d'une compromission de machine, ou du propriétaire de la machine lorsqu'il est victime.

Il faut donc une approche en deux temps pour l'investigation numérique. La première va être la phase de collecte des informations sur la machine en indexant la totalité des données contenues. Cette opération peut être réalisée par DFF précédemment évoqué. Ensuite, suivant le type d'artefact remonté, il va falloir contextualiser l'information pour lui donner un sens et comprendre son origine et d'une façon plus générale, ce qui l'entoure. OSINT va donc permettre de collecter l'ensemble de ces informations, afin de les indexer sur les sources ouvertes disponibles sur Internet.

5. OSINT

OSINT (Open Source INTelligence) est une méthodologie de renseignement et d'investigation visant à rechercher, collecter et indexer tous types d'informations non classifiées (au sens militaire/renseignement du terme) à partir de sources dites « ouvertes » (i.e. d'accès public). L'obtention de ces informations n'est pas assimilable à un vol de données confidentielles ou publiques. Les sources exploitées vont être les médias, Internet, les journaux, les livres, etc. Plus généralement, toutes sources pouvant être vecteur d'informations.

Le but étant de cartographier une cible humaine et/ou technique afin d'en déterminer le degré de menace potentielle. Cette technique a vu le jour au début des années 2000 lorsque les renseignements américains ont dû faire face à une nouvelle menace qu'ils ne connaissaient pas (les terroristes islamistes). Ils ont dû comprendre la nature de la menace, son degré et comment s'en défendre. Ils ont dû collecter l'ensemble des informations disponibles relatives à cette menace (en plus des services de renseignements) afin de pouvoir mieux l'appréhender.

Ainsi, cette démarche à permis d'adapter « les techniques de renseignement » à la SSI, d'une part grâce à la partie « information gathering » et/ou « network discovery » et d'autre part grâce à la partie investigation numérique.

L'atout majeur de ces techniques réside dans la faible interaction qu'il peut y avoir directement avec le système d'information de la cible. Puisque l'ensemble des requêtes va se faire sur des sources ouvertes.

De plus, les interactions avec le système d'information cible seront de même natures que celles engendrées par un internaute « classique ». Le but étant de simuler à la fois le même type de trafic que ce soit au niveau des métriques (bande passante, temps entre deux requêtes, fréquence des requêtes) que la nature même du trafic (requêtes HTTP/DNS normales, pas de malformations volontaires, pas d'injections de quelques natures que ce soit). L'objectif étant de recueillir le plus d'informations possible. Une fois cette cartographie établie, le second objectif est de déterminer l'exposition de la cible ainsi que sa vulnérabilité. Du fait de la faible interaction avec le système d'information, la furtivité est un atout considérable et le point central de l'ensemble de ces techniques.

Dans un premier temps, on va définir ce que l'on entend par « source ouverte en SSI » en détaillant l'utilisation de chacune de ces sources, puis dans un second temps on fera

un état de l'art autour de ces techniques. Enfin, on expliquera les raisons qui nous ont poussées à développer notre propre framework, on détaillera son architecture, son fonctionnement ainsi que ses finalités associées.

5.1 Définition d'une source ouverte en SSI

Une source ouverte en SSI se compose de plusieurs types d'informations récupérables sur Internet :

5.1.1 Les moteurs de recherches

Les moteurs de recherche auxquels nous allons nous intéresser sont les suivants : Google, Yahoo, Bing et Shodan. L'idée est l'utilisation des fonctionnalités avancées des différents moteurs en utilisant ce que l'on va appeler des « dorks ». Une « dork » est l'utilisation d'un mot clé pour cibler plus spécifiquement une recherche.

Exemples :

- « site**:** 'toto.com' », les différents moteurs vont retourner l'ensemble des sites web appartenant à toto.com (sous domaines inclus)
- « filext**:** sql », les moteurs de recherche vont retourner l'ensemble des fichiers indexés, dont l'extension est « sql ». Dans ce cas typique, l'intérêt réside dans le fait qu'il est possible de trouver dans ces fichiers, des hashes de mots de passe pouvant être soumis à des dictionnaires pour « cassage », voire des mots de passe en clair.

Chaque moteur de recherche a ses propres mots clés. Par exemple, sur le moteur « bing », le mot clé « ip » permet de renvoyer l'ensemble des sites hébergés par l'adresse IP spécifiée. Au final, c'est une sorte de reverse DNS utile et peu couteux à mettre en place. Une autre « dork » plutôt utile sur le moteur « shodan » est « org » qui retourne l'ensemble des résultats liés à l'organisation spécifiée qu'il aura réussi à identifier.

On voit bien que l'utilisation ingénieuse de ces différentes « dorks » va nous permettre de mieux cibler le type d'informations recherchées. Il est alors possible de trouver de cette manière des serveurs vulnérables et/ou déjà compromis. Il existe une liste exhaustive des « google dorks » [15]

5.1.2 Les réseaux sociaux

Les réseaux sociaux vont nous permettre de croiser différentes informations. Par exemple, dans le cas de Facebook nous allons pouvoir soumettre des adresses mails récupérées, afin de savoir si un compte lui est affilié ou non. L'intérêt majeur de cette démarche est de pouvoir obtenir une identité réelle à partir d'un simple mail. Ainsi, dans le cadre d'investigations numériques ou d'attaques ciblées, Facebook devient vite une très bonne source d'informations puisque par ce biais nous pouvons établir des interactions entre individus ou groupe d'individus, voire récupérer des informations de géolocalisation.

Parallèlement, Twitter, va nous permettre d'obtenir des informations similaires.

Dans le cas des réseaux sociaux tels que Viadeo/Linkdin, ce sont les informations professionnelles qui vont pouvoir être exploitées.

5.1.3 Les Whois IP/Domaines

Les IP Whois vont nous permettre d'avoir des détails sur l'environnement de la cible que nous souhaitons cartographier. À partir de ces informations nous allons déterminer où cette cible est la plus présente en terme de géolocalisation et en terme de réseaux à la fois sur les plages IP rencontrées, mais aussi des informations BGP comme les AS. Cela nous permet de contextualiser où se trouve la cible. Une fois ces informations trouvées ; il faut les croiser avec des bases de réputation (Team Cymru, ShadowServer) ou utiliser des outils de ranking comme celui développé par Alexandre Dulaunoy [16].

5.1.4 Les registrars

Les registrars vont nous donner des informations sur la personne morale détentrice du domaine, avec parfois des adresses postales ou mails. Ces informations vont venir en compléments de ce qui a été abordé dans le paragraphe précédent.

5.1.5 Les DNS

Le but des actions sur les DNS va être de nous fournir des informations sur les domaines voire les sous domaines associés. A l'aide de dictionnaires, on pourra découvrir des sous domaines, et peut-être déterminer si d'autres services que le web sont accessibles (smtp, ftp ...).

Dans le cas où les reverse DNS sont configurés, on pourra obtenir des informations complémentaires. De plus si le DNS est mal configuré, et que les requêtes sur le transfert de zone sont acceptées, alors il sera possible d'obtenir l'ensemble des informations que le DNS a sous son autorité.

5.1.6 Les agrégateurs

Les agrégateurs tels que Robtext, le RIPE regroupent une quantité significative d'informations liées aux domaines, sans qu'il soit nécessaire de devoir les retester. Il suffit de donc de les interroger afin d'obtenir l'ensemble des informations précédemment évoquées.

5.1.7 Les certificats SSL

Les certificats SSL regorgent souvent d'informations qui peuvent nous être utiles. Tout d'abord, si le SSL est configuré sur un service, la connexion SSL va se faire quelle que soit la requête par IP ou par Vhost. Nous avons donc le certificat qui nous est fourni. Et dans la plupart des

cas, il y a des adresses mails et surtout les Vhosts sur lesquels le certificat est censé être utilisé pour le handshake SSL.

Le cas le plus riche est le cas des wildcard : dans le champ CN du certificat, nous avons l'information de type *.toto.com et dans un autre champ du certificat (OtherDNSNames), il y a l'ensemble des Vhosts pour lesquels le certificat peut être utilisé sans engendrer une erreur SSL.

5.1.8 Le code source de l'applicatif web

L'intérêt d'analyser le code source d'une application web (Fingerprinting, comparaison) est de comprendre l'environnement que nous sommes en train de cibler, de découvrir éventuellement un premier niveau de vulnérabilités si l'on fait de l'offensif, et de récupérer davantage d'informations dans le cadre d'une investigation afin de pouvoir procéder à la mise sous scellé.

5.1.9 L'état de l'art

Il existe de nombreux outils qui permettent en partie la récupération d'informations en sources ouvertes. Le plus connu est Maltego dont le modèle d'affichage des données, bien qu'esthétique, devient vite illisible au fur et à mesure que la quantité de données à afficher augmente.

De plus, Maltego est basé sur une architecture client/serveur. La partie serveur est située dans le datacenter de Paterva en Afrique du Sud, et depuis peu, il est possible d'héberger la partie serveur dans son propre datacenter, mais le coût devient vite élevé. De la même façon, les fonctionnalités les plus intéressantes de Maltego sont payantes et le code est « close source ». Il est néanmoins possible de coder des extensions (appelées transformations) mais cela reste évidemment limité face à un logiciel complètement Open Source.

Le concurrent Open Source de Maltego est Netglub, ce dernier souffre du même problème que Maltego au niveau des performances.

Enfin, il existe des scripts alternatifs tels qu'esearch ou theharvester qui, soit ne fonctionnent pas, soit de par sa composition monolithique ne permet quasiment pas d'ajout de fonctionnalités.

C'est la raison pour laquelle, nous avons souhaité développer un framework Open Source.

6. Développement d'OSINT

Ce framework a été développé avec les fonctionnalités suivantes :

- Recherche sur les différents moteurs de recherche avec randomisation des requêtes
- Recherche DNS, PasteBin, réseaux sociaux, Whois, certificats, bannières...
- Analyseur lexicométrique et syntaxique
- Outils utilisés pour l'analyse de discours
- Analyse factorielle des correspondances
- Classification hiérarchique
- Analyse de contenus
- Utilisation de bases de données d'images
- Analyse de liens
- Fingerprinting d'applications web
- Scoring sur les liens et les images
- Utilisation de screenshots pour faciliter le tri
- Géolocalisation des sites web
- Classement des données

La structure du framework peut-être représentée sous forme modulaire, comme le montre la figure 1.

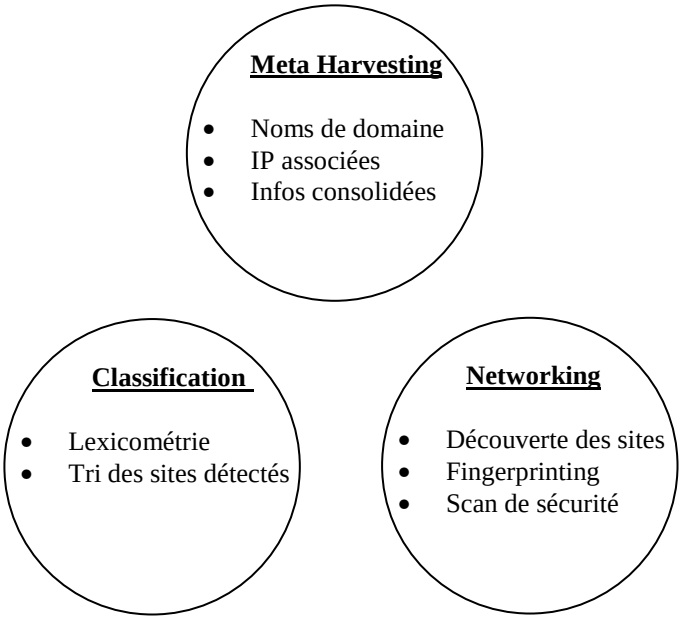

Fig. 1 : structure du framework OSINT

Ci-dessous, la liste des outils et scripts qui ont été utilisés pour le framework :

- Python
- MongoDB/redis
- PhantomJS/CasperJS
- Nmap/NSE
- SSLyse patché
- TheHarvester patché

- BlindElephant/whatweb
- Iramuteq/R
- Maxmind
- OVI
- Amazon EC2

7. Finalités et méthodologies

Au final, le framework répond à plusieurs besoins :

- Avoir une cartographie de la cible la plus complète possible
- Identifier les pays dans lesquels les sites sont hébergés
- Faire un inventaire des sites web
- Connaître l'environnement de la cible

Afin d'utiliser le framework efficacement, il est nécessaire de respecter une méthodologie :

1. Avec les metaharvester, il faut lancer les recherches ciblées (en utilisant les dorks appropriées pour chaque moteur cf. 5.1.1)
2. Faire une géolocalisation de chaque site
3. Faire des screenshots
4. Récupérer les métadonnées et les index de l'ensemble des sites trouvés
5. Faire une analyse lexicométriques pour trouver de nouveaux axes de recherches
6. Faire le tri avec les différents screenshots des informations retournées
7. Sortir l'ensemble des réseaux/hébergeurs sur lesquels les sites ont été trouvés
8. Lancer le scanner sur ces réseaux en fonction de la répartition des sites
9. Effectuer un screenshot de l'ensemble des Vhosts remontés
10. Refaire un tri
11. Relancer le metaharvesting de l'étape 1 pour effectuer une passe supplémentaire, en tenant compte des données déjà collectées et triées. En fonction des résultats attendus/obtenus, on pourra limiter le nombre de passes nécessaires.

    Cette méthodologie pourra faire l'objet d'un atelier proposant l'utilisation du framework.

8. Exemples de tests

Afin de valider la méthodologie, ainsi que les résultats retournés, on va cartographier les sites gouvernementaux français (gouv.fr). En résultat, nous obtenons un résultat de 583 sites en 4 passes. Le résultat est disponible sous forme d'un fichier texte [17].

Parallèlement, l'ensemble des sites web de sociétés privées a été soumis au test, néanmoins, les résultats obtenus ne sont pas publics, mais la quantité de données n'a pas posé de problèmes de traitements particuliers.

9. Références

[1] Haskell, Neal H. (2006). « The Science of Forensic Entomology » in *Forensic Science and Law: Investigative Applications in Criminal, Civil, and Family Justice*, 431–440. Edited by Cyril H. Wecht and John T. Rago. Boca Raton: CRC Press, an imprint of Taylor and Francis Group. ISBN 0-8493-1970-6. Page 432.

[2] E. Casey. *Digital evidence and computer crime: forensic science, computers and the Internet*. Academic Pr, 2004. ISBN 0121631044.

[3] Dan Farmer and Wiest Venema, *Forensic Discovery*, ISBN-10 020163497X

[4] McDorman, Richard E. (2010). *Liberty and Scientific Evidence in the Courtroom: Daubert v. Merrell Dow Pharmaceuticals, Inc. and the New Role of Scientific Evidence in the Criminal Courts*. ISBN 978-0-9839112-2-7.

[5] Bernstein, David Eliot, *Frye, Frye, Again: The Past, Present, and Future of the General Acceptance Test* (2001). George Mason Law & Economics Research Paper No. 01-07.

[6] Brian Carrier, *Open Source Digital Forensics Tools, The Legal Argument* , http://www.digital-evidence.org/parpers/opensrc_legal.pdf

[7] http://www.wibu.com/en/hacker-contest.html

[8] http://www.x-ways.net/winhex/manual.pdf

[9] Tim Newsham, Chris Palmer, Alex Stamos, Jesse Burns, *Breaking Forensics Software : weaknesses in Critical Evidence Collection* , iSEC Partners, Inc.

[10]http://git.digital-forensic.org

[11]http://lists.digital-forensic.org

[12]http://tracker.digital-forensic.org

[13]http://www.dfrws.org/2010/challenge/index.shtml

[14]Johannes Stuttgen, *Selective imaging : Creating efficient forensic Images by selecting Content First*, 2011, http://www1.cs.fau.de/filepool/thesis/diplomarbeit-2011-stuettgen.pdf

[15]http://www.exploit-db.com/google-dorks/

[16]https://github.com/adulau/bgpranking-API

[17]https://github.com/sebdraven/OSINT/blob/master/processing/gouv_domaine.txt