# Compte-rendu de réunion

Référence RÉSIST/2012-05

15 mai 2012

## Table des matières



## 1 Revue d'actualité

Etienne Maynier a fait une courte revue de l'actualité des dernières semaines en matière de sécurité informatique. Parmi les éléments qu'il a relevés :

- Le botnet *Flashback* qui a touché les machines Apple, avec près de 550 000 ordinateurs concernés.
- Des incidents de sécurité, par exemple Global Payments (1,5 millions de numéros de carte bancaires volés) ou Twitter (55 000 comptes et mots de passe publiés sur Pastebin).
- Les relations complexes entretenues avec la Chine et les entreprises chinoises en matière d'informatique et d'appels d'offres publics.

## 2 Codenomicon, the story of how I hacked into your TV

La présentation de M. Rikke Kuipers a commencé par un bref rappel des techniques de *fuzzing* et de l'intérêt de ce type d'approche : cela fournit une forme d'assurance sécurité pour les matériels et les logiciels. Le fuzzing autorise une approche de type « relever un dysfonctionnement/le comprendre/le corriger ». L'automatisation des tests permet une exploration plus large de la surface d'exposition du matériel ou du logiciel. Le fuzzing permet de détecter relativement rapidement des situations de déni de service (redémarrage intempestif, blocage complet), de dégradation des performances, d'instabilité des systèmes, voire des vulnérabilités.

Note

Chaque test positif suppose ensuite une analyse de l'incident, afin d'en déterminer les raisons et de les corriger. Il n'y a pas toujours de vulnérabilités exploitables au-delà d'un déni de service.

Il existe schématiquement plusieurs façons de mettre en œuvre du fuzzing :

**aléatoire** C'est la méthode la plus élémentaire. Sur un flux valide, l'outil modifie au hasard des bits ou octets. Une telle façon de procéder rencontre rapidement ses limites sur des flux structurés (échange protocolaire, etc.) : une modification dans la structure des données peut les rendre invalides et provoquer leur rejet très en amont du code à tester.

**par blocs** L'outil dispose d'une certaine capacité d'analyse de bas niveau. Il peut identifier des structures de base (entiers, chaînes de caractères) et agir dessus. Cela permet notamment d'envoyer des informations « structurellement correctes » qui contiennent des données sur lesquelles l'outil a apporté des modifications.

**par modèle** L'outil met complètement en œuvre la RFC ou le protocole ciblé, en y injectant des données anormales. La compréhension du protocole spécialise l'outil (ou un module de celui-ci). Elle permet cependant de couvrir l'ensemble du dit protocole d'échange, y compris des sous-cas spécifiques qui ne peuvent être atteints que suite à des chemins logiques complexes.

**par interception** Plutôt que mettre en œuvre une RFC ou un protocole, cette méthode repose sur une interception des échanges entre deux composants. Il est ensuite nécessaire d'analyser ces échanges pour relever ce qui pourrait être modifié. Cette méthode se révèle moins efficace que la précédente car, par nature, elle est limitée aux échanges qui ont pu être observés.

L'utilisation du fuzzing a progressé dans le temps. Dans les années 1990, les méthodes aléatoires sont devenues relativement populaires. Sur la fin de la décennie, c'est le tour des modèles par blocs. Au début des années 2000, les fabricants de matériels réseau commencent à utiliser de façon fréquente le fuzzing pour tester leurs équipements. Au milieu de la décennie 2000, les opérateurs de télécommunications ont commencé à inclure ce genre de tests dans leurs critères d'acceptation du matériel.

Les équipements électroniques modernes présentent des fonctionnalités très étendues. Ainsi, une télévision dernier cri peut disposer du Wifi et du Bluetooth, de ports USB, d'une connexion à Internet. Elle est capable de décoder de très nombreux formats de fichiers (images, sons, vidéos), dispose d'un navigateur Web embarqué, etc. La surface d'accès à un téléphone évolué couvre des centaines d'interfaces, de protocoles et de formats de fichiers.

Les équipements mobiles pouvant être, au cours d'une même journée, connectés à de nombreux réseaux différents, cette très importante surface d'accès ne demande qu'à être exploitée. Un appareil embarqué peut être une cible intéressante pour peu qu'il soit joignable par Wifi ou Bluetooth. Un équipement de type NAS doit, par nature, accepter de nombreux protocoles différents (FTP, Smb, Nfs...) et être très accessible sur le réseau.

Deux vidéos ont été présentées :

1. Un fuzzing sur un téléviseur récent, en demandant la lecture de vidéos « spécialement formattées ». La vidéo permet de constater qu'à certains moments, la télévision redémarre toute seule. On peut facilement imaginer la conception d'une vidéo (qui pourrait être diffusée via Youtube ou tout autre service équivalent, voire dans une campagne de publicité à une heure de forte écoute) dont l'affichage sur un certain type de téléviseur provoquerait le blocage de celui-ci.
2. Un fuzzing sur des appareils embarqués dans des voitures haut de gamme. Les équipements testés acceptent le Bluetooth. La vidéo montre qu'ils sont très vulnérables, allant jusqu'à nécessiter un retour usine pour être remis en fonction.

Une démonstration en direct sur un boîtier routeur/wifi a montré que, en injectant des informations via SIP, il est trivial de faire redémarrer le boîtier.

Dans chacune de ces démonstrations, la conséquence minimale est un déni de service sur l'équipement visé. Il est cependant concevable d'analyser l'incident pour déterminer si une exploitation plus fine serait possible, avec une prise de contrôle de la cible et un rebond vers le réseau intérieur. Dans le cas d'un véhicule, cela pourrait signifier un accès aux systèmes de commande.

## 3 Wallix, traçabilité des administrateurs

M. Marc Balasko a présenté les outils de traçabilité des administrateurs développés par Wallix.

La question de la traçabilité découle souvent de questions relatives à une conformité règlementaire (Pci-DSS, Arjel, e-Santé, etc.). Les problématiques sont connues : gestion des mots de passe, traçabilité des actions, authentification, etc. Concernant les administrateurs, les droits d'accès étendus dont ils diposent ajoutent une composante au problème.

Plusieurs points constituent souvent des pierres d'achoppement :

1. la gestion des mots de passe (rotation, robustesse, non-partage, non-recyclage, etc.),
2. le cycle de vie des comptes (nominatifs, comme il se doit),
3. l'analyse *a posteriori* d'une erreur et son imputabilité à une personne,

4. la gestion des prestataires, notamment s'ils disposent d'accès externes à des équipements critiques.

Wallix propose un système de type « bastion d'administration », qui constitue un point de passage obligé pour la connexion sur certains systèmes. Cela permet de répondre aux questions précédentes :

- Les administrateurs disposent de comptes nominatifs sur le bastion, à partir duquel ils peuvent rebondir vers les équipements concernés. Ces rebonds sont contrôlés (liste de cibles autorisées).
- L'imputabilité est assurée par la journalisation des connexions entrantes et sortantes et leur attribution à un utilisateur du bastion.
- La traçabilité est possible par l'enregistrement (texte et/ou vidéo) des sessions de travail
- Les mots de passe sur les systèmes cibles peuvent être changés automatiquement par le bastion (éventuellement tous les jours ou toutes les heures).

Plusieurs questions ont concerné le risque de laisser un outil changer les mots de passe des administrateurs sur des équipements critiques, notamment en cas de défaillance de l'outil en question. La solution mise en œuvre par Wallix est, dès lors que l'on désire activer cette fonctionnalité, de demander une clé GPG publique ainsi qu'une adresse électronique. Les mots de passe définis par l'outil, chiffrés par la clé publique fournie, sont envoyés à cette adresse électronique.

La gestion des sessions d'administration à l'aide de clients lourds demande une modification de l'architecture logique. Au lieu d'avoir un client lourd sur le poste de l'administrateur, avec lequel il se connecte sur une machine cible, le client est déporté sur un serveur de type TSE. L'accès à ce serveur TSE se fait au travers du bastion (donc avec enregistrement de la session, imputabilité, etc.), et les clients lourds sont ensuite lancés pour accéder aux systèmes visés.