
# Compte rendu de la conférence *hack.lu* 2007

par Jérôme LEONARD (jerome.leonard@hapsis.fr) et Saâd KADHI (saad.kadhi@hapsis.fr)

Cette conférence s'est déroulée sur 3 jours avec une première journée dédiée aux workshops, les deux autres journées étant consacrées à des présentations et "lightning talks" (présentations improvisées). Beaucoup de sujets intéressants ont été abordés, dont un particulièrement couvert, et ce sous différents angles : les malwares. L'exploitation des mécanismes de SAP, l'importance du développement sécurisé des applications Web ou encore les faiblesses des implémentations Wifi sont autant de sujets qui ont aussi attiré notre attention. Enfin, tout aussi intéressant mais hors de nos domaines d'intérêt habituels, les mécanismes de sécurités des E-passports et des transmissions RDS-TMC destinés aux messages d'information en temps réel pour les systèmes de navigation embarqués (SatNav) ont été abordés...ainsi que leurs faiblesses.

## Définitions

**Bot Herder**
Personne qui collectionne ou contrôle les botnets.

**Botnet**
réseau de PCs infectés et contrôlés par des utilisateurs malicieux via un serveur C&C (command & Control).

**C&C**
PC ou réseau de PC permettant d'envoyer des commandes aux botnets.

## Analyse forensic des botnets - FCCU belge (Federal Computer Crime Unit)

Après avoir présenté Nepenthes[1], un outil de type "honeypot" permettant de capturer des malwares en simulant des vulnérabilités Windows, Christophe Monniez et Miguel Blauwbloeme ont présenté leur méthodologie d'analyse statique et dynamique des malwares recoltés.

L'objectif principal de la FCCU n'est pas de comprendre les mécanismes ni les caractéristiques unitaires des malwares, mais plutôt de trouver quelle en est la source, ce qu'ils font sur le poste de travail, où se connectent-ils et quelles sont les données partagées ?

Pour capturer rapidement des malwares, il est possible d'utiliser Nepenthes. Il s'agit d'un outil passif, simulant des vulnérabilités connues et qui capture les malwares qui tentent de les exploiter.

### Capturer des malwares avec Nepenthes

Cet outil est une sorte de honeypot (pot de miel) et peut être positionné à plusieurs niveaux dans une architecture :

- En frontal sur internet (entre le réseau interne et Internet),
- Sur une DMZ internet,
- Sur un réseau intranet afin de traquer la diffusion de malwares en interne.

**DDoS**
Distributed Denial of Service (Déni de Service Distribué).

**Enregistrement A (DNS)**
Enregistrement désignant une adresse ; correspondance entre une adresse IP et un nom d'hôte.

**Enregistrement NS (DNS)**
Enregistrement désignant un serveur de nom.

**Fuzzing**
Technique consistent à injecter des données aléatoires dans les entrées d'un programme. Si le programme échoue (en crashant ou en générant une erreur) alors des défauts sont présents dans celui-ci. Ceci permet de détecter rapidement des erreurs de programmation triviales, telles que les erreurs sur les vérifications d'entrées/sorties. (source Wikipedia).

**IRC**
Protocole utilisé pour discuter en temps réél basé sur une architecture client/serveur. (ancètre de la messagerie instantanée telle qu'on la connait aujourd'hui).

**Mothership**
Serveur central regroupant tous les services nécessaires à un service Fast-Flux.

**Parseur/Parser**
Abus de langage, définissant un outil d'analyse syntaxique.

**TTL *(Time To Live - DNS)***
Temps déterminant la durée valable d'un enregistrement DNS (association d'une adresse IP à un nom donné).

Les malwares téléchargés sont stockés dans un répertoire particulier suivant leur provenance (adresse IP) et leur MD5.

Il est possible de synchroniser Nepenthes avec des *sandboxes* (conteneurs virtuels où les programmes peuvent évoluer dans un environnement protégé ce qui en permet l'analyse) tel que Norman Sandbox ou Sunbelt Sandbox. D'autres outils de type honeypot (honeytrap, honeyd ...) sont aussi utiles dans la capture de malwares.

### Analyse statique et dynamique (suivant les informations recherchées)

L'analyse doit permettre de découvrir un maximum d'informations sur les sites distants et la source des malwares.

Pour ce faire, deux approches sont possibles : l'analyse statique, et l'analyse dynamique, le tout, réalisé dans un environnement virtualisé et déconnecté d'internet.

L'analyse montre que ces programmes tentent de se connecter sur des serveurs IRC distants (Internet Relay Chat, "ancêtre" de la messagerie instantanée). Les postes infectés, aussi appelés "zombies" se connectent, à l'insu de l'utilisateur, sur ces serveurs de chat et attendent les instructions.

Cette analyse permet de découvrir des informations intéressantes telles que des noms de domaines (par extension, à qui appartiennent ces noms de domaines et les adresses IPs), les ports de communications, les commandes de contrôle ou les services ouverts sur le poste de travail infecté. Ces données seront utilisées pour rechercher les auteurs de ces programmes malicieux.

## Botnets & Botherders (Botnets et Collectionneurs de bot) - Hilar Leoste de Shadowserver.org

La FCCU travaille en collaboration avec la fondation shadowserver[2], un groupe de professionnels de la sécurité qui étudie, partage, traque et rapporte sur l'activité des malwares de manière générale.

### Construction du réseau

L'infection d'un PC par un malware peut avoir plusieurs origines :

- D'abord par l'exploitation de vulnérabilités ; les malwares sont capables de s'auto-propager sur demande des botherders,
- Par la messagerie (instantanée et mail) ; les utilisateurs ne se doutent pas qu'un programme malicieux peut s'exécuter à leur insu lors de l'ouverture d'une pièce jointe dont la source n'est pas sûre,
- Lors de la visite de pages web, au gré des redirections, par l'intermédiaire des partages P2P ...

Les PCs infectés (appelés zombies) se connectent à un serveur C&C dans l'attente d'ordre. L'ensemble représente un botnet. La communication entre les zombies et les serveurs C&C se fait par IRC, ou HTTP (certains botnets plus complexes fonctionnent en P2P).

### Un botnet pour quels usages ?

- Attaques par DDoS,
- Auto-propagation,
- Spamming,
- Écoutes de trafic réseau et contournement des tunnels chiffrés,
- Écoutes de frappes clavier ...

Diverses utilisations peuvent être envisagées suivant les besoins et les motivations (espionnage industriel, réseaux parallèles, sponsorisation d'état ...).

### Comment s'en défendre ?

La "traque" des botherders et le démantèlement des botnets se révèle être une affaire plutôt complexe. D'abord, les botherders utilisent aussi des mécanismes de défense comment l'utilisation d'enregistrement DNS court (TTL très court) et dynamique, la redirection vers d'autres serveurs C&C, ou le chiffrement des communications qui rend l'analyse difficile.

### In real life ...

Un botnet (SDBot) capturé en août 2006 a utilisé 32 noms d'hôtes différents pour les serveurs C&C en moins de 60 jours, a changé de serveur au moins 40 fois, et au moins 6 autres moyens de communication sont connus pour ce nom d'hôte. Le serveur a diffusé 51 binaires uniques dans ces 2 mois, 2/3 étaient des variantes de SDBot. Les mécanismes de diffusions sont classiques utilisant des vulnérabilités connues sur Windows. Pendant ces 2 mois, ce botnet a été utilisé pour lancer des DDoS (souvent des groupes ennemis comme cible), récupérer des IDs et mots de passe (keylogging), pour de la fraude internet ...

**XSS**
*(Cross-Site Scripting)* : vulnérabilité rencontrée dans les applications web permettant d'injecter du code dans les pages appelées par d'autres utilisateurs.

**XSRF**
(*Cross-site Request Forgery*) : vulnérabilité permettant de mener une attaque dont l'objectif est de transmettre des commandes non autorisées à un site, à l'insu d'un utilisateur authentifié. Attaque aussi connue sous le nom de *One-Click Attack*.

**Zombie/Drone**
PC compromis qui reçoit des commandes via un serveur C&C.

### Ce qui nous attend

Le futur des botnets est assuré et leur champ d'action ne demande qu'à s'élargir. On peut tout à fait envisager, si cela n'est pas déjà le cas, que des botnets soient utilisés pour installer des rootkits, capables de détecter des antivirus, pour gagner des privilèges ... De nouveaux protocoles de communication font déjà leur apparition (VOIP, IM, P2P, HTTP), il ne manque plus que le chiffrement (pour bientôt ? ). On peut s'attendre aussi à des botnets plus petits, distribués et mieux protégés, et financés par des "industries" du crime.

Aujourd'hui on commence à trouver des outils simples d'utilisation basés sur HTTP permettant de contrôler à distance un botnet (Black Energy, Zunker, MPack, ICEPack, Barracuda, Pinch ...).

### Les mesures de Prévention

Certaines mesures de sécurité peuvent prévenir et/ou détecter l'émergence d'un botnet au sein d'un réseau local ; Il est d'abord clairement recommandé de suivre les guides de sécurisation des systèmes d'exploitation et des services, ainsi que de limiter les privilèges des utilisateurs; concernant le trafic réseau, il est fortement recommandé de faire passer la navigation web des utilisateurs à travers un proxy, d'enregistrer l'ensemble des requêtes et de les analyser. enfin, la sensibilisation des utilisateurs n'est pas à négliger.

## La fin du principe de défense en profondeur ? Quid des antivirus ? - Thierry Zoler & Sergio Alvarez [3]

Le principe de défense en profondeur est une technique d'origine militaire qui consiste à placer plusieurs mécanismes de sécurité entre un attaquant potentiel et le système cible. Typiquement, et conformément aux guides pratiques de sécurité, l'architecture antivirale d'un réseau d'entreprise est mise en oeuvre suivant cette logique.

Cette présentation se propose de remettre en cause quelques mythes :

- Un antivirus est une solution sécurisée et rend les systèmes et les réseaux sécurisés,
- Les antivirus sont développés par des experts en sécurité,
- "J'ai un antivirus, mon système est sécurisé, je ne serai pas infecté",
- "Mon antivirus détecte aussi les virus qui ne sont pas encore connus".

Dans les faits, un antivirus est développé par des programmeurs comme tout autre logiciel, possède une surface d'attaque d'autant plus grande qu'il est capable d'analyser un très grand nombre (quelques centaines) de types de fichiers tels que .doc, .zip, .exe, .sh ... Par ailleurs, aujourd'hui, sur un réseau, il est nécessaire d'avoir un antivirus pour quasiment tous les services : un pour les systèmes, un pour la messagerie, un pour les serveurs SQL, un pour les serveurs de fichiers, un pour les proxies ... Par conséquent, les antivirus offrent une surface d'attaque proportionnelle au nombre de moteurs différents, au nombre d'analyseurs sur une infrastructure.

Ce principe a été illustré avec l'évasion des antivirus par un mail. Un message provevant d'internet est analysé par au moins trois moteurs antiviraux : un première fois sur une passerelle MX, une seconde sur le serveur de messagerie interne, et une troisième fois sur le poste de travail du destinataire ; trois analyses, trois moteurs différents et autant de *parseurs*.

Un problème majeur concernant un grand nombre d'antivirus est que le fonctionnement par défaut laisse passer les mails contenant les pièces jointes qu'ils n'arrivent pas à analyser. Ce comportement permet de se concentrer sur les mécanismes d'évasion afin de permettre à une pièce jointe d'atteindre la cible voulue ; l'objectif étant de contourner ou de prendre le contrôle de l'antivirus. D'ailleurs, la démonstration aura montré qu'il est possible de contourner un antivirus en modifiant une caractéristique mineure du format ZIP, ou en ajoutant du texte dans un fichier au format PE (Portable Executable). Ces comportements n'ont pas été prévus par les éditeurs, par conséquent l'antivirus n'est pas capable d'analyser ces fichiers, et les laisse passer.

Les impacts potentiels sont très variés, contournement de la détection, DDoS, modification de la configuration, élévation de privilèges, exécution de code à distance ...surtout que les moteurs sont souvent réutilisés dans les IDS/IPS ...

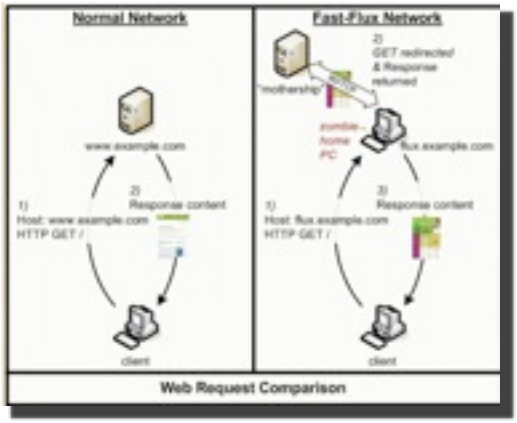
Ceci met en évidence que l'interprétation du principe de défense en profondeur appliqué à l'architecture antivirale d'un réseau d'entreprise n'est pas adaptée. Même si la multiplication d'antivirus permet de réduire la fenêtre d'exposition à une infection virale, des effets de bords peuvent apparaître et doivent être pris en compte. Ces services de désinfection virale doivent aussi être cloisonnés et sécurisés.

## Références

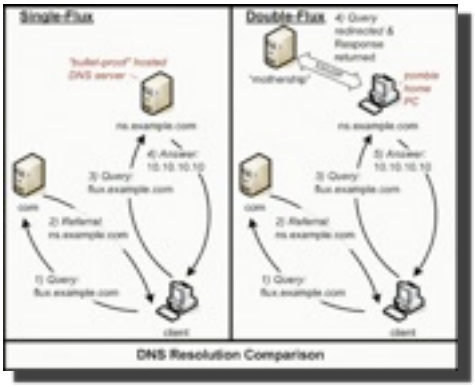

1. http://nepenthes.mwcollect.org/
2. http://www.shadowserver.org/
3. http://www.nruns.org
4. http://savannah.nongnu.org/projects/requestrodeo/


## Fast-Flux - Lance Spitzner

La seconde journée a débuté avec un speech de Lance Spitzner concernant les réseaux Fast-Flux. Ces réseaux sont une nouvelle étape dans l'amélioration des réseaux des infrastructures criminelles. Un réseau de services *Fast-Flux* est un réseau de machines compromises possédant des enregistrements DNS constamment modifiés, parfois en l'espace de quelques minutes. L'objectif de Fast-Flux est d'avoir plusieurs enregistrements IP(des centaines ou des milliers) pour un même nom DNS (par exemple *www.exemple.com*). Les machines infectées par des malwares sont transformées en reverse-proxies et redirigent les flux vers un serveur central ou *mothership*. Ainsi, avec le jeu autours des noms DNS, la véritable adresse IP vers laquelle un client est attiré varie d'une fois sur l'autre, mais au final, c'est le même serveur qui répond.




Plus sophistiqués encore, et plus complexes, les réseaux *Double-Flux* ajoutent un niveau supplémentaire en terme de redondance. En effet, non seulement les *enregistrements A* changent fréquemment, mais aussi les *enregistrements NS*. Par conséquent, dans ce cas là, l'interrogation DNS d'un client est envoyé au serveur DNS faisant autorité au moment de la requête, qui transfert la requête au *mothership*. Ce serveur DNS faisant autorité au moment de la requête n'est autre qu'un zombie, qui agit comme un reverse-proxy et transfert la requête au *mothership* qui fourni l'adresse d'un autre zombie actif à ce même moment et capable de gérer les communications web.




De tels mécanismes présentent par exemple l'intérêt d'ajouter des couches de protection entre les victimes et un site de phishing. Par conséquent, le site en question peut difficilement être traqué et retrouvé.

Les moyens de prévention sont faibles ; l'analyse des malwares peut permettre de découvrir les *motherships*, et la surveillance des flux HTTP et DNS anormaux peut permettre de découvrir les systèmes infectés.

Ces services sont souvent offerts par les RBN (*Russian Business Network).* Ce sont des organisations criminelles dont les revenus se font essentiellement sur les *cybercrimes* tels que ceux précédemment mentionnés. Leurs organisations sont difficilement traçables ; à cause des aspects techniques avancés utilisés d'une part, et des facilités administratives permettant l'enregistrement des compagnies et des domaines d'autre part.

## Exploiter les fonctionnalités de SAP - Mariano Nunez di Croce

SAP repose sur la possibilité de communiquer et d'interagir avec d'autres systèmes externes. Pour cela, il existe l'interface RFC ou Remote Function Call (dont l'origine remonte aux années 1980) qui permet d'appeler une fonction sur un système distant ; une fonction pour être accessible à distance doit être définie comme telle (taggée, Remote-Enable). Les appels de fonctions distantes se font par l'intermédiaire d'une passerelle de service qui gère les communications des systèmes SAP entre eux, et avec d'autres systèmes externes.

L'analyse de l'implémentation de l'interface RFC a permis de découvrir certains aspects intéressants. Par écoute du trafic, il est possible d'obtenir les informations d'authentification, les fonctions appelées ainsi que les paramètres et leur contenu, des informations sur les tables et leurs contenues, et autres informations sur les clients et les serveurs. D'autre part, certaines fonctions installées par défaut, permettent d'obtenir des informations sur les implémentations distantes :

- fonctions installées,
- disponibilité du systèmes SAP distant,

## Présentations

**Analyse des botnets**
http://hack.lu/pres/Malware%20Analysis.odp

**Botnets & Botherders**
http://hack.lu/pres/shadowservers-hl2007.ppt

**Défense en profondeur**
http://www.nruns.com/ps/The_Death_of_AV_Defense_in_Depth-Revisiting_Anti-Virus_Software.pdf

**Fast-Flux**
http://www.honeynet.org/speaking/index.html

**Exploiter les fonctionnalités SAP**
http://www.cybsec.com/upload/Hack.lu07-Attacking_The_Giants_SAP-v2.pdf

**Bugs dans les implémentations wifi**
Une présentation similaire a été donnée au SSTIC :
http://actes.sstic.org/SSTIC07/WiFi_Fuzzing/SSTIC07-Butti_Tinnes-WiFi_Fuzzing.pdf

**Compromission et durcissement des applications web**
http://hack.lu/pres/dhanjani-hack.lu-2007.pdf

**Sécurité du RDS-TMC**
http://hack.lu/pres/hacklu2007_barisani_bianco.pdf

- informations concernant le systèmes distant (Version du noyau SAP, nom d'hôte, moteurs de base de données, adresse du moteur de base de données, système d'exploitation utilisé,
- ...

En matière de sécurité, il n'existe pas de mécanisme d'authentification pour interroger un système distant. Par ailleurs, certaines fonctions font l'objet de vulnérabilités de type débordement de tampon et permettent notamment de passer des commandes à distance.

La démonstration prévue a été l'occasion pour le présentateur de présenter *Sapyto,* le premier framework publique permettant de réaliser des tests d'intrusion sur une infrastructure SAP. Cette démonstration a permis de démontrer avec quelle facilité il est possible de réaliser des attaques de type *Man In The Middle*, de se faire passer pour un serveur externe auprès de la passerelle SAP et d'attaquer le serveur SAP.

## Bugs dans les implémentations Wifi : de nouvelles vulnérabilités - Frank Veysset, Laurent Butti, Julien Tinnes

La démonstration proposée par la cellule R&D de Orange présente la mise en évidence de plusieurs vulnérabilités découvertes dans les implémentations wifi de différents point d'accès. Cette recherche de vulnérabilités s'est faite principalement en utilisant la technique du *fuzzing*.

les implémentations 801.11 (wifi) sont de plus en plus complexes, et par conséquent contiennent de plus en plus de code et donc plus de vulnérabilités potentielles.

À ce jour, plusieurs failles ont été découvertes sur des implémentations connues du grand public ; majoritairement des problèmes de longueur de champs dans les trames ont été découverts dans des drivers Netgear et D-Link, sans oublier MadWifi (utilisé sur les systèmes GNU/Linux et certains points d'accès). D'autres vulnérabilités ont été découvertes dans la gestion de l'EAP avec WPA/WPA2 sur les points d'accès Cisco. Ces faiblesses peuvent permettre, sous certaines conditions, d'exécuter des commandes sur le système distant du point d'accès.

Ces problèmes d'implémentation représentent un risque non négligeable. Les mécanismes de sécurité présents aujourd'hui sur les réseaux wifi (WPA ou WPA2) n'offrent aucune protection contres des attaques sur les drivers.

## Compromission et durcissement des applications web - Nitesh Dhajani

La complexité des applications web ne cessant de s'accroître, et par conséquent, la part de faiblesses potentielles dans le code, le présentateur commence par rappeler l'importance de leur sécurisation.

D'après Gartner, la très grande majorité des attaques d'aujourd'hui, cible des applications web. Ceci est est du à deux facteurs principaux :

- Les applications web sont composées de millions de lignes de code, ce qui augmente leur surface d'attaque,
- Les protections et contrôles sur le réseau n'arrêtent pas les attaques sur les applications.

Aujourd'hui, quand les réseaux sont cloisonnés, protégés par des pare-feux, et quand la "webisation" des applications est le modèle ou le standard de sécurité pour les entreprises, celles-ci représentent une cible de tout premier choix.

Les principaux vecteurs d'attaques sont les failles XSS et XSRF présentes dans les applications.

### XSS

Les vulnérabilités XSS sont les plus populaires. Ces faiblesses sont dues à une défaillance dans la validation du code généré par les applications en fonction des informations entrées par l'utilisateur. L'exemple trivial est une application web qui est appelée par une URL contenant une variable entrée par un utilisateur ; La page, résultant du traitement de cette variable par l'application, est envoyée à l'utilisateur sans passer pas des procédures de validation ; par conséquent le navigateur de l'utilisateur reçoit le contenu de la variable tel que l'utilisateur l'a remplie. C'est là qu'il est possible d'injecter du code qui sera interprété par le navigateur, dont le résultat sera imposé à l'utilisateur.

Concrètement, considérons l'appel de la page :

```
http://www.site.com/appli.cgi?name=BOB
```

cet appel permet d'afficher "BOB" sur la page. Sans traitement du contenu de la variable "name", l'appel de la page :

```
http://www.site.com/appli.cgi?name=<SCRIPT>alert('xss');</
script>
```

va ouvrir une popup inscrivant “xss”. Ceci est inoffensif, mais le code injecté peut être beaucoup plus agressif et complètement transparent aux yeux des utilisateurs ; on pourrait envisager de faire en sorte qu’un utilisateur envoit des informations personnelles vers un autre site (Cookie de session, informations d’authentification, historique du navigateur ...).

Plusieurs mécanismes peuvent permettre de se prémunir de cette faille :

- valider le contenu des pages envoyées au client en *échappant* certains caractères. Par exemple pour la requête précédente, voici ce qui devrait être renvoyé au navigateur (les caractères < et > sont *échappés* en **<** et **>**):

```
&lt;script&gt;alert('xss');&lt;/script&gt
```

- envisager de faire de la validation des entrées dans les applications,
- avoir une approche par liste blanche concernant les entrées de données par les utilisateurs.

Pour autant, ces mécanismes ne protégeront pas à 100% contre toutes les failles XSS : des failles persistantes peuvent provenir des données stockées dans les bases de données.

### XSRF

Les attaques de type XSRF (ou CSRF) ciblent l’utilisateur en exploitant la confiance d’un site dans un navigateur ayant établi une session. L’objectif est de faire exécuter des commandes sur un site à l’insu de l’utilisateur. Par exemple :

```
<IMG SRC=”http://www.somebank.com/
transaction.cgi?
amount=9999999&to_account=1234567890” />
```

Plusieurs solutions permettent de prévenir ce type d’attaque :

- Ne pas compter sur le *Referer* pour s’assurer que l’utilisateur est bien à l’origine de la demande,
- Ne pas compter sur le fait que la méthode POST soit utilisée pour opérer les commandes de l’utilisateur,
- Utiliser des marqueurs aléatoires à inclure dans les requêtes provenant de l’utilisateur,
- Coté navigateur, un projet, RequestRodeo[4], ayant pour objectif le développement d’un proxy ou modules permettant de se prémunir de ces attaques à été initié il y a un an.

La combinaison de XSS et de XSRF peut s’avérer particulièrement efficace et permettre d’utiliser le navigateur web comme proxy vers un intranet. On peut envisager une attaque XSRF qui permettrait d’exploiter une faille XSS d’un site intranet, et d’y d’injecter du code.

## Divers

Deux autres présentations/démonstrations ont fait impression : l’E-passeport et la sécurité des communications RDS-TMC.

### Le E-passeport - Zubair Khan

Les passeports électroniques sont aujourd’hui utilisés dans plusieurs régions dont l’Europe pour le contrôle d’identité. Ces nouveaux passeports contiennent une puce RFID dont la sécurité, de la puce elle-même et des mécanismes d’utilisations est remise en cause.

En terme de sécurité, les puces RFID souffrent de certaines faiblesses :

- elles sont détectables à distance,
- il est possible d’intercepter leurs communications avec un lecteur,
- elles sont clonables,
- elles peuvent être infectées par des virus.

Pour la vérification d’un passeport, certains mécanismes obligatoires permettent d’authentifier le lecteur et d’établir une clé de session empêchant ainsi l’écoute à distance de l'interrogation des passeport. Cependant, cela n’empêche pas le clonage. D’autres mécanismes basés sur une infrastructure à paire de clés permet de s’assurer de la validité de la puce RFID embarquée dans un passeport. Cependant, ces mécanismes ne sont pas obligatoires dans les processus de validation d’un passeport.

La démonstration proposée par le présentateur aura permis de montrer la “simplicité” avec laquelle la protection en lecture seule d’une puce RFID a pu être cassée pour autoriser l’écriture.

Les risques associés sont réels tels que le vol d'identité ou l'utilisation d'une puce comme déclencheur d'un action à distance (utilisation en tant que détonateur par exemple).

### Les communications RDS-TMC - Daniele Bianco, Andrea Barisani

Si un prix avait du être attribué pour la présentation la plus captivante et amusante, certainement que celle-ci aurait été en compétition sinon récompensée.

Le RDS-TMC est un protocole radio destiné à transporter des informations sur le trafic routier à destination des dispositifs de navigation par satellite embarqués dans les véhicules récents. Le fait est, que ce protocole peut être détourné.

Après avoir présenté comment ils ont construits leurs propres émetteurs et récepteurs pour étudier/analyser le fonctionnement de ce protocole, les présentateurs ont mis en évidence les faiblesses du RDS-TMC et leurs conséquences.

Certains dispositifs de navigation embarqués utilisent le protocole RDS-TMC pour informer l'utilisateur sur les conditions de circulation en temps réel (bouchons, météo, taux de remplissage d'un parking ...). C'est un protocole radio que tout le monde reçoit. Après avoir réussi à décoder et comprendre les mécanismes internes de ce protocole, les présentateurs ont réussi à construire un transmetteur et à émettre des informations. Ces informations ont pu être interprétées par le système de navigation embarqué de leur véhicule de test. Aucun dispositif d'authentification n'a été nécessaire pour que les messages soient acceptés, interprétés par le systèmes et proposé au conducteur.

Une démonstration a été proposée : elle a suggéré au système un incident sur un itinéraire préalablement choisi par l'utilisateur et a déclenché un détournement de celui-ci pour arriver à la destination choisie.

Les conséquences de tells faiblesses sont importantes. À titre d'exemple voici les incidents qui peuvent être soumis aux conducteurs sans aucun moyen d'authentification ni d'intégrité :

- accidents, bouchons, conditions météorologiques factices ; le système demande à l'utilisateur s'il souhaite un contournement du problème.
- Fermetures de routes, ponts tunnels ; le système calcule en silence un nouvel itinéraire.

D'autre part, les tables d'événements pré-enregistrées dans le système de navigation étudié, contiennent un certain nombre de messages d'alertes relatives à la sécurité tels que : incident terroriste, combat de taureaux, raid aérien, crash aérien, alerte à la bombe ... (sick!)

Bien que le risque associé ne soit pas gigantesque, le fait est que l'utilisation de tels systèmes est de plus en plus répandue. Pourtant, ces problématiques ne sont pas ou peu prises en compte par les constructeurs.