Véhicules Connectés et Cybersécurité

Abstract

The hundreds of millions of cars on our roads every day is a unique source of highly valuable data. While technically, connected car already makes those data available outside the vehicle, this export raises serious questions on how to secure the process. Connected cars raise premium challenges for the integrity of vehicle global security. It also raises issues about who collects and owns those data or how the process may respect users privacy. This talk exposes the ongoing work inside AGL (Automotive Grade Linux) to secure the global architecture from data collection to their export on the cloud. How to run untrusted applications without compromising your security, how to push only selected data from the car to the cloud, how to separate real time operations from less critical ones as entertainment, this without exploding development cost or limiting developers creativity.

Auteurs

Fulup Le Foll, fulup@iot.bzh

Stéphane Desneux, sdx@iot.bzh

Version 1.0 - octobre 2016

Contexte Économique

Aujourd'hui, tous les analystes s'accordent à dire que l'Internet des Objets est déjà passé devant le « Big Data[1] » en termes de potentiel de marché. On dénombrait 20 milliards d'objets connectés fin 2015 ; ils devraient être près de 50 milliards dès 2020. Coté automobile, on prévoit 200 à 250 millions de véhicules connectés pour 2020. Les analystes prédisent un quadruplement du marché dans les cinq prochaines années. Ceci représente un ajout en valeur aux marchés automobiles existants de 150 milliards de dollars. En 2018, le marché des services connectés aux véhicules devrait déjà atteindre 40 milliards d'euros, et va continuer à croître jusqu'à la généralisation des véhicules autonomes vers 2035/2040.

Les changements en cours représentent pour le marché automobile une révolution aussi importante que l'arrivée d'internet dans le secteur bancaire, ou la mise en place de la gratuité de la voix chez les opérateurs téléphoniques. Dans cette révolution à venir, il y aura des gagnants et des perdants, comme dans toutes les révolutions : il est donc critique pour les grands constructeurs de ne pas rater la marche. Pour ce faire, ils se doivent d'acquérir des technologies et des compétences qu'ils n'ont pas, et notamment celles issues de la téléphonie mobile pour la partie acquisition et du Cloud et du « Big Data » pour les traitements.

Toutefois, il est important de rappeler qu'une voiture n'est pas une TV ou un téléphone. Il est donc nécessaire de reformater les technologies existantes afin de les rendre compatibles avec les contraintes de l'automobile. Tout le monde comprend qu'une voiture doit être mieux sécurisée qu'une télévision, ou que la durée de vie moyenne d'un véhicule est de 20 ans alors que celle d'un téléphone mobile dépasse rarement 3 ans.

Le challenge de l'automobile, comme celui de tous les autres marchés de l'internet des objets, est triple :

- réussir à simplifier la mise en œuvre d'un modèle de cybersécurité applicatif de bout en bout
- fournir un ensemble d'outils qui rendent acceptable le coût et la complexité attachés au développement et à la mise en œuvre d'un projet en environnement hautement sécurisé
- assurer que le modèle de sécurité couvre les applications sur l'ensemble de leurs cycles de vie

1 https://fr.wikipedia.org/wiki/Big_data

La situation actuelle

Jusqu'à présent, les véhicules comme la grande majorité des autres infrastructures industrielles n'étaient que faiblement connectés à Internet : le risque d'attaques cybercriminelles reste ainsi trop souvent considéré comme faible. À de très rares exceptions près, le cyber-risque n'est pas considéré comme un élément structurant des architectures systèmes. Trop souvent les méthodes utilisées se limitent à un « disaster recovery plan », une prévention des dénis de services, un schéma de lutte contre les virus et autres malwares.

Avec 250 millions de véhicules connectés sur les routes dès 2020, il est évident que les business d'attaque des voitures et autres systèmes industriels vont devenir économiquement viables très rapidement. Combien une société de transports serait-elle prête à payer en voyant ses camions ne démarrant pas un lundi matin ? Quel serait le manque à gagner pour un patron pêcheur dont les traces et waypoints disparaîtraient suite à une mise à jour de son système de cartographie, ou d'un navire marchant dont le GPS/AIS le positionnerait à 500m de sa véritable position[2] ?

Les récentes attaques ont montré que les systèmes actuels restent relativement simples à attaquer. À ce jour (octobre 2016), Tesla est le dernier à avoir fait la une de la presse avec l'attaque de ses voitures par un groupe de hackers Chinois[3]. Toutefois même s'ils essaient d'en minimiser la portée médiatique, la liste des constructeurs dont la cybersécurité a été mise en faute est longue. BMW a dû rappeler 2.2 millions de véhicules ; l'attaque des Jeeps durant l'été 2015 a coûté plus de 150 millions de dollars à Fiat-Chrysler. De nouveaux cas plus ou moins importants continueront à apparaître aussi régulièrement que les grandes marées sur les côtes Bretonnes : là où il y a un business, il y a toujours des fournisseurs.

L'attaque des Jeeps a cependant marqué un point de rupture, et ce n'est que depuis début 2016 que le risque cybercriminel est vraiment pris en compte comme un élément structurant des architectures automobiles. Avant 2016, la tendance était de considérer que puisque la cybersécurité était sous-traitée à un intégrateur qui en assurait l'entière responsabilité, les constructeurs n'étaient pas vraiment concernés. Au final, trop de monde pensait que l'attaque d'un véhicule ou d'un système industriel était si complexe, si longue et si chère que personne ne serait intéressé. Aujourd'hui, nous savons que ce n'est pas le cas. Les hackers et autres « chapeaux noirs[4] » sont très intelligents, très bien formés et ont le temps pour eux. Enfin, il y a suffisamment d'argent en jeu pour qu'ils trouvent facilement les financements leur permettant de mener à bien leur besogne.

2 https://www.transportshaker-wavestone.com/cybersecurite-maritime-faut-il-attendre-un-naufrage/
3 http://www.01net.com/actualites/des-hackers-ont-reussi-a-pirater-une-tesla-model-s-a-distance-1039190.html
4 https://fr.wikipedia.org/wiki/Black_hat

Aujourd'hui, toute l'industrie admet que les voitures de demain, comme presque tous les systèmes industriels, seront connectés à Internet. Tous ces systèmes seront régulièrement attaqués et certaines attaques seront nécessairement couronnées de succès. Il faut donc mettre en place les mécanismes pour réduire les surfaces d'attaque et installer des défenses pour se protéger des risques connus, mais aussi prévoir des systèmes de mise à jour automatiques pour corriger les erreurs et se prémunir des attaques encore inconnues. L'age moyen d'une voiture en Europe est de 9.5 ans pour une durée de vie de 20 ans. Personne n'est en mesure de prévoir les technologies de cyberattaques qui seront disponibles dans 10 ou 20 ans : permettre la mise à jour du système n'est donc pas une option.

Pourquoi AGL (Automotive Grade Linux) ?

Comme rappelé initialement, les changements en cours représentent une révolution pour le marché automobile. Il est donc critique pour les grands constructeurs de ne pas rater la marche. Le risque pour les constructeurs automobiles de subir avec la voiture connectée le même sort que celui réservé à Nokia[5] avec le Smartphone est loin d'être nul.

Pour éviter ce scénario catastrophe, les constructeurs ont de nombreux challenges à surmonter :

- **Réduire le coût du logiciel :**

  À titre d'exemple, le coût facturé au client pour le système électronique d'une Mercedes Class E n'a augmenté que de 1650€ entre 2010 et 2015, alors que sur la même période, le coût pour le constructeur augmentait de 6500€[6].

  Si à court terme, cet écart reste acceptable sur les véhicules commercialisés entre 60 et 70k€, il est évident que d'une part il n'est pas soutenable sur les véhicules de gammes inférieures et que même sur le haut de gamme sur le long terme, les constructeurs devront réduire leurs coûts afin de continuer à innover sans faire exploser les prix.

  Dans le bas de gamme, le problème est encore plus critique : il faut compter 650€ pour un système de navigation standard intégré dans la voiture, alors que l'équivalent n'est facturé que 150€ par TomTom pour un système externe équivalent.

  Si les constructeurs ne veulent pas perdre le marché des véhicules connectés, ils doivent drastiquement réduire le coût de leur électronique embarquée.

5 https://www.linkedin.com/pulse/nokia-failure-story-ibrahim-abd-elaziz?forceNoSplash=true
6 http://www.strategyand.pwc.com/media/file/Connected-Car-Study-2015.pdf

- **Ne pas perdre la guerre du « Big Data » :**

  Dans les 50 milliards de dollars ajoutés au marché du « Big Data » par les véhicules connectés, une part non négligeable provient de la fourniture de services : le streaming de musique, la mise à jour de la cartographie, la gestion du trafic, etc. Les constructeurs sont dans l'obligation de trouver comment récupérer une partie des sommes attachées à la fourniture de ces services, car seule une petite part peut être facturée au client lors de l'achat du véhicule.

  Dans cette lutte, les constructeurs ont un point fort : eux seuls contrôlent les données fournies par leurs voitures.

  Toutefois ils ont aussi beaucoup de faiblesses : peu ou pas de connaissance au business du « Big Data » ; pour certains d'entre eux, une santé financière qui n'a rien d'exceptionnel et enfin le temps qui joue contre eux.

  À l'inverse, leur principal concurrent « Google », qui espère récupérer 30% de ce marché en devenir, a des budgets quasi-illimités. Il contrôle environ 80 % des téléphones mobiles, possède une cartographie mondiale ainsi que de nombreux autres services très utiles et qu'une grande majorité d'utilisateurs aimerait voir intégrés dans les voitures.

- **Aller vers la voiture autonome :**

  Même si la voiture autonome n'est pas pour demain, tout le monde s'y prépare activement. La quantité d'innovations nécessaires pour atteindre ce Graal est sans précédent dans l'histoire de l'automobile.

  Malheureusement, les plateformes actuelles ne sont absolument pas compatibles avec le rythme d'innovation effréné nécessaire à la progression vers des véhicules 100 % autonomes. Aujourd'hui, aucun constructeur ne possède des systèmes évolutifs et réutilisables qui lui permettraient de véritablement capitaliser l'expérience d'un véhicule à l'autre, d'une génération à la suivante.

- **Passer les freins sociologiques :**

  Les études montrent que même si les utilisateurs sont demandeurs de nouvelles fonctionnalités, ils ont aussi des craintes non seulement sur la cybercriminalité, mais aussi sur la protection de leur vie privée.

Comme le montre l'étude de McKinsey[7], la relation entre voitures connectées et consommateurs est complexe : la majorité des consommateurs ne maîtrisent que très partiellement les enjeux attachés aux avantages/inconvénients des véhicules connectés et pourtant demandent de plus en plus de fonctionnalités qui imposent de connecter leurs véhicules à Internet.

Ils veulent plus de confort de conduite, avec des cartes mises à jour automatiquement ou des listes de musiques synchronisées sur leurs préférences personnelles depuis leur domicile ou leur téléphone. Ils demandent plus de sûreté avec une détection automatique des écarts de route, des piétons, des distances de sécurité, etc.

Toutefois, dans le même temps, ils craignent pour le respect de leur vie privée et restent sceptiques sur la capacité des constructeurs à les protéger des attaques cybercriminelles.

Tant que les voitures connectées resteront destinées à des passionnés de technologie, il n'y aura que très peu de risques. En revanche, avant d'équiper monsieur Tout-le-monde, il faudra être certain que la technologie ne décevra pas, sous peine de voir un rejet en masse de plusieurs années par le marché.

Afin de dépasser un usage limité au véhicule de luxe et atteindre le marché de masse, les constructeurs doivent réduire les coûts des systèmes informatiques embarqués de manière drastique. Aucun constructeur n'étant assez puissant pour résoudre seul ce problème, ils ont dû se résoudre à développer des solutions communes afin de distribuer les coûts de recherche et développement entre différents acteurs de la filière. L'objectif des constructeurs est, tout comme dans la téléphonie, d'avoir une plate-forme partagée unique qui supporte tous les composants non visibles du client. Une fois les services de base assurés par cette plate-forme commune à bas coût, ils pourront focaliser leur effort financier sur la partie « expérience utilisateur » et les autres éléments qui fournissent de véritables différenciateurs sur leur marché.

AGL est l'un des principaux consortiums mondiaux travaillant sur le sujet; il regroupe tous les constructeurs Japonais mais aussi les grands intégrateurs mondiaux comme Continental ou Panasonic. S'il existe d'autres consortiums comme Genivi[8] en Europe, AGL est aujourd'hui celui qui est le plus avancé dans la fourniture d'une plateforme logicielle à destination des développeurs automobiles. C'est aussi la seule plateforme automobile qui intègre de manière native un modèle de cybersécurité de bout en bout.

7 http://www.mckinsey.com/industries/automotive-and-assembly/our-insights/whats-driving-the-connected-car

8 https://www.genivi.org/

Cybersécurité et Options Techniques

Garantir la source des logiciels

Sur un site sensible, la première des mesures de sécurité consiste à vérifier l'identité des personnes. De la même manière sur un système informatique, la première des sécurités consiste à vérifier la provenance du logiciel. Authentifier la source d'un programme est la seule solution pour garantir que vous allez bien exécuter le code que vous pensez vouloir exécuter.

Signatures et PKI

Bien que les techniques de PKI[9] soient connues depuis de nombreuses années, elles restent complexes à mettre en œuvre. Il faut une autorité pour créer les clefs, les distribuer et les révoquer. Il faut également trouver un système qui puisse protéger les véhicules à titre individuel, sans toutefois tomber dans une complexité trop grande qui serait impossible à gérer. Bien que le marché de l'automobile soit minuscule comparé à celui des téléphones, il faut tout de même compter une centaine de millions de nouveau véhicules chaque année. De plus, la durée de vie (20 ans) ainsi que le nombre d'intervenants (fabriquant, sous-traitants, mécaniciens, propriétaires, locataires ou simples utilisateurs) rendent la gestion de la PKI bien plus complexe que dans une entreprise traditionnelle.

Avant le démarrage

A l'inverse des systèmes d'entreprise qui sont statiques et dont l'accès peut au moins en théorie être régulé, il est impossible d'interdire l'accès physique aux systèmes embarqués d'un véhicule. Il faut donc configurer l'électronique d'une manière telle qu'elle ne puisse pas être compromise, même par quelqu'un ayant un accès matériel au dispositif :

- Supprimer tous les mécanismes de bas niveau comme le JTAG qui pourrait être utilisé pour injecter du code malicieux directement en mémoire.
- Garantir que le « bootloader » ne peut pas être compromis, par exemple en l'inscrivant dans une mémoire en lecture seule, directement dans le composant en usine.
- Mettre en place des mécanismes de fusibles physiques, qui une fois « grillés » interdisent l'accès à toutes les fonctions de développement, de debug, etc.

À noter que toutes les protections doivent rester compatibles avec les procédures de maintenance, dont certaines peuvent imposer la réinitialisation totale du système dans un garage à partir d'un périphérique physique comme une clef USB ou une « valise » d'intervention.

9 https://fr.wikipedia.org/wiki/Infrastructure_%C3%A0_cl%C3%A9s_publiques

Pendant le démarrage

La première des choses est de garantir qu'on exécute le bon noyau système avec les bonnes options de lancement. Cette technique appelée « boot sécurisé », bien qu'un peu complexe à mettre en œuvre, est disponible sur toutes les électroniques embarquées destinées à la production.

Tous les composants critiques du système doivent être vérifiés par des clefs de signatures cryptographiques. De plus, Linux intègre un système appelé IMA[10] pour la mesure de l'intégrité de l'architecture et un autre appelé EVM pour module de vérification étendue. Ce dernier vérifie que les attributs étendus des fichiers qui contiennent entre autres les droits et privilèges n'ont pas étés modifiés de manière accidentelle ou malicieuse.

Outre la complexité de mise en œuvre qui doit rester acceptable pour ne pas faire exploser les coûts, les voitures ont un problème supplémentaire attaché à la vitesse de démarrage initial. A titre d'exemple, la vidéo de la caméra de recul doit pouvoir s'afficher seulement 2s après la mise du contact : il faut donc un système de démarrage non seulement sécurisé mais aussi très rapide.

Après le démarrage

Une fois que le système a effectivement démarré, il faut pouvoir vérifier que les applications et services sont bien ceux que l'on pense qu'ils sont. Si les services restent assez souvent groupés avec le reste du système d'exploitation et peuvent donc être ajoutés aux vérifications IMA/EVM, c'est plus rarement le cas pour les applications qui ont en général un cycle de vie indépendant du système de base.

Pour les applications le système doit :

- **Vérifier le code** : L'application doit être téléchargée dans un paquet scellé qui contient non seulement l'application, mais aussi ses règles et privilèges de sécurité. L'ensemble doit être protégé par des signatures cryptographiques qui garantissent l'intégrité et l'origine.
- **Gérer les dépendances** : Installer une application impose presque toujours d'en vérifier les dépendances. Celles-ci doivent être disponibles et accessibles avec le niveau de privilèges attaché à l'application, sous peine d'annuler son installation ou sa mise à jour.
- **Démarrer une application/service** : avant le lancement de tout processus, son intégrité doit être certifiée. Afin de limiter l'impact de ce contrôle sur les performances, il est indispensable d'utiliser les coprocesseurs présents sur la plateforme matérielle mais aussi d'implémenter des mécanismes de tampons pour limiter le nombre de calculs de signatures d'intégrité.

10 https://wiki.gentoo.org/wiki/Integrity_Measurement_Architecture

Garantir la source d'un composant et en assurer l'intégrité tout en restant capable de le mettre à jour pendant plus de 10 ans est un véritable challenge. Les expériences passées ont montré que perdre une clé pouvait compromettre globalement le système sur le très long terme. Il faut donc mettre en place un système qui soit à la fois souple, performant mais, une fois encore, dont la complexité reste suffisamment masquée afin de ne pas faire exploser les coûts.

Isoler et Compartimenter

Comme dans le monde réel, la sécurité des logiciels est implémentée en isolant les contextes d'exécution. Dans le monde physique, on construit des grands murs avec des portes bien gardées ; dans le monde virtuel, on peut aussi utiliser des mécanismes d'isolations pour garantir qu'un contexte d'exécution ne puisse pas « déborder » chez son voisin.

- **Virtualisation** : Elle peut soit être « hardware » et utiliser les caractéristiques avancées du processeur associé à un hyperviseur du type XEN/KVM ; ou alors comme dans le cas de Docker être purement logicielle et s'appuyer sur les mécanismes standards du noyau Linux comme les NameSpaces ou les Cgroups.
- **Les contrôles d'accès** : Historiquement, tous les dérivés d'Unix supportent les contrôles dit « discretionary » où le propriétaire d'une ressource décide de qui peut accéder, modifier ou utiliser celle-ci. Les versions plus récentes de Linux ont ajouté la notion de contrôles « obligatoires » où les règles d'accès ne sont plus fixées par le propriétaire de la ressource, mais par un administrateur de sécurité qui fixe de manière globale les règles d'accès.
- **Les restrictions de permissions** : elles peuvent être fixées pour un utilisateur donné, ou par application. Exemples : « un jeune conducteur n'a pas le droit de conduire à plus de 90km/h » ou « l'application Autoradio n'a pas le droit d'accéder à la camera de recul ».

L'ensemble de ces techniques sont complémentaires : elles doivent toutes être combinées pour arriver à un système à la fois flexible, performant et dont le modèle de sécurité reste maintenable sur le long terme.

Le modèle de Cybersécurité d'AGL

Le projet AGL a adopté le modèle de sécurité de bout en bout proposé par IoT.bzh. L'annonce en a été faite lors de l'Automotive Linux Summit de Tokyo en Juillet 2016, et l'intégration dans les référentiels d'AGL est disponible depuis la version 2.0[11].

Bien que le modèle de sécurité d'AGL soit encore en phase de développement, il propose dès à présent un certain nombre de composants techniques qui permettent de commencer les développements.

Garantir l'origine du code

- **Boot sécurisé** : valorise les capacités des coprocesseurs de cryptographie pour accélérer le démarrage. Un mécanisme souple de gestion des clés pour supporter les petites séries et les modes de développement.
- **Vérification des applications** : Les applications sont gérées via des paquetages au format « widget » (.wgt) du W3C. Lors de l'installation, le système vérifie non seulement l'origine, mais aussi que les permissions demandées sont en corrélation avec la source du logiciel.
- **Gestion du consentement** : la plupart des applications nécessitent des permissions et il est important que l'utilisateur puisse conserver le contrôle de la plateforme, ce tout particulièrement pour les données qui concernent sa vie privée.

Une Architecture en Couches

Le modèle d'AGL est basé sur une architecture en couches avec un modèle qui autorise l'exécution d'applications « non-trustées ». Le modèle de sécurité applicative s'appuie d'une part sur les mécanismes du noyau de Linux, comme les CGroups[12] ou les NameSpaces[13] ainsi que sur le mécanisme SMACK[14] pour le contrôle obligatoire des accès. D'autre part, la gestion de la communication inter-applications passe par un mécanisme de « binder[15] » qui effectue la vérification des permissions attachées aux APIs[16] à l'aide de la base de privilèges Cynara[17], sur le même principe que Tizen[18] chez Samsung.

11 https://gerrit.automotivelinux.org/gerrit
12 https://fr.wikipedia.org/wiki/Cgroups
13 https://en.wikipedia.org/wiki/Linux_namespaces
14 https://fr.wikipedia.org/wiki/Simplified_Mandatory_Access_Control_Kernel
15 http://docs.iot.bzh/docs/architecture/en/dev/reference/ap/binder/afb-overview.html
16 https://fr.wikipedia.org/wiki/API
17 https://wiki.tizen.org/wiki/Security:Cynara
18 https://www.tizen.org/

Une architecture distribuée

Le modèle de développement d'applications AGL s'appuie sur une transparence d'API qui permet au développeur d'écrire ses composants de manière réutilisable sans avoir à se soucier de l'architecture finale de déploiement.

Par exemple un agent d'acquisition d'un bus CAN[19] sera coupé en deux parties : d'une part un module traitant le niveau bas qui va récupérer les messages binaires et les interpréter afin de les rendre compréhensibles aux applications ; d'autre part, un module « logique business » qui est en charge d'implémenter la logique applicative (ex : signal « le véhicule bouge »).

Une fois développé, chaque module est comparable à une brique de Lego. Il peut être intégré soit dans un processus unique, soit sur des processus indépendants ou même avoir une partie exécutée dans la voiture et l'autre dans un service de type « cloud » quelque part sur Internet.

Évidement, chaque modèle d'implémentation a un impact sur la sécurité comme sur les performances ; toutefois il est important de comprendre que ce n'est plus au développeur de décider du modèle de déploiement. Chaque module développé devient réutilisable dans des architectures différentes qui peuvent par exemple correspondre à différents niveaux de fonctionnalités et de prix des véhicules concernés.

Une vérification statique de la sécurité

Une voiture connectée compte plus de 100 millions de lignes de code[20], avec une majorité de composants développés par des équipes externes sur lesquelles les constructeurs n'ont absolument aucun contrôle. Prétendre qu'il est possible d'utiliser les systèmes traditionnels de certification pour un tel assemblage tient au mieux de l'ignorance, au pire de la malhonnêteté intellectuelle. La seule solution est donc de restreindre par l'extérieur les capacités de nuisance des applications comme des services.

AGL 2.0 intègre un mécanisme de manifeste où chaque application déclare les ressources qu'elle compte utiliser ainsi que les APIs qu'elle expose. Ces demandes sont ensuite comparées aux permissions attachées au niveau de signature et de contrôle d'origine du composant concerné. Dans le cas où les demandes outrepassent ces autorisations, l'application n'est pas installée. En revanche, si elles sont acceptables, l'application est installée et la base Cynara de protection des APIs ainsi que les labels SMACK et CGroups sont provisionnés afin de garantir que l'application ne pourra jamais outrepasser les droits qui lui ont été attribués.

19 https://fr.wikipedia.org/wiki/Controller_Area_Network
20 https://www.technologyreview.com/s/508231/many-cars-have-a-hundred-million-lines-of-code/

Mécanismes de cybersécurité en cours d'intégration

Bien qu'il soit déjà utilisable, le système de cybersécurité d'AGL est loin d'être finalisé. Beaucoup de travaux sont en cours, et parmi eux certains sont plus avancés et ont de bonne chance d'être intégrés pour la fin 2016 :

- **Gestion de la Trusted Zone** : Ce système spécifique à la famille de processeurs ARM implémente un sous-système sécurisé qui assure une isolation « électronique » entre la partie « trustée »[21] de la partie système d'exploitation.
- **Hypervision** : Les processeurs sont de plus en plus puissants et afin de réduire les coûts, il faut regrouper les fonctions. Toutefois, pour faire cohabiter des systèmes avec des niveaux de sécurité ou de certification différents sans craindre que l'un puisse déteindre sur l'autre, il faut impérativement implémenter une isolation proche du niveau « électronique » pour garantir une isolation parfaite entre chacun des sous-systèmes. Pour ce faire, on utilise des hyperviseurs et notamment XEN[22] ou KVM[23] qui sont des produits opensource disponibles en standard sur toutes les plateformes Linux. Historiquement utilisés dans le monde des serveurs, les hyperviseurs sont à présent disponibles également dans des formes adaptées aux systèmes embarqués.
- **Identité de l'utilisateur** : Beaucoup d'usages attachés aux services de « Big Data » nécessitent de vérifier l'identité de l'utilisateur. A titre d'exemple : sans savoir qui conduit le véhicule, le système ne peut pas fournir une liste des musiques préférées de l'utilisateur ou encore valider le paiement automatique d'un parcmètre. La gestion d'identité est aussi indispensable aux nouveaux modes de consommation des véhicules, notamment pour assurer la gestion des véhicules partagés.
- **Mise à jour « on the air »** : Aussi appelé SOTA[24], ce mécanisme est partiellement disponible dans AGL 2.0. Toutefois un travail d'intégration avec le système de développement comme avec le modèle de sécurité reste à faire avant qu'il puisse être classé comme composant natif d'AGL. La mise à jour automatique est un élément critique de la politique de cybersécurité de tout système connecté à Internet.

21 https://en.wikipedia.org/wiki/Trusted_system
22 https://www.linux.com/news/xen-virtualization-takes-automotive
23 http://www.linux-kvm.org
24 http://events.linuxfoundation.org/sites/events/files/slides/OTA%20Updates%20in%20AGL%20Using%20OSTree.pdf

Conclusion

Comme pour l'ensemble de l'industrie connectée, le travail de sécurisation des véhicules de demain ne fait que commencer. À ce jour, les attaques connues sont toutes les faits de « white hats[25] » et les véritables modèles d'attaques malicieuses restent à découvrir.

Les attaques cybercriminelles peuvent être comparées à un incendie : si on intervient suffisamment vite, elles sont rarement graves. Avec le déploiement de centaines de millions de véhicules connectés, les premières véritables attaques ne devraient pas tarder. Il sera alors essentiel d'apprendre rapidement et de réagir dans les délais les plus brefs avant que l'ensemble du système ne s'embrase.

Comme toujours en sécurité, le diable se cache dans le détail. Avoir les bonnes briques de cybersécurité n'est pas suffisant : il faut s'assurer qu'elles soient parfaitement intégrées avec un niveau de complexité qui reste acceptable. Faire un système sécurisé simple est sans aucun doute le plus grand challenge des architectes de cybersécurité. Un système trop complexe n'a aucune chance d'être accepté : il ne serait pas maintenable, les utilisateurs trouveraient toujours un moyen de le contourner et le marché refuserait de le financer.

Les véhicules de demain seront tous connectés et tous les véhicules connectés seront attaqués. Le risque est connu mais la bonne nouvelle est que les technologies pour le minimiser sont disponibles et matures. Construire une architecture sécurisée pour les automobiles connectées n'est pas forcement simple, mais reste néanmoins réalisable. Si l'industrie accepte de joindre ses efforts afin de trouver les financements compatibles avec la hauteur des enjeux, alors les véhicules connectés seront fiables et les constructeurs garderont le contrôle des données de leurs utilisateurs. Dans le cas contraire, il est probable qu'un « Google » prendra ce marché et réduira la valeur ajoutée des constructeurs de la même manière qu'il a réduit celle des fabricants de téléphones.

25 https://fr.wikipedia.org/wiki/White_hat

Références

Internet des Objets

https://fr.wikipedia.org/wiki/Internet_des_objets

https://en.wikipedia.org/wiki/Internet_of_Things

Big Data / Mégadonnées

https://fr.wikipedia.org/wiki/Big_data

Analyses

http://iot-analytics.com/iot-market-forecasts-overview/

http://www.gartner.com/newsroom/id/2819918

http://www.forbes.com/sites/gilpress/2014/08/18/its-official-the-internet-of-things-takes-over-big-data-as-the-most-hyped-technology/

Tizen

https://www.tizen.org/

https://en.wikipedia.org/wiki/Tizen

https://www.tizen.org/blogs/bdub/2015/tizen-mwc-iot-growing-and-so-tizen

Automotive Grade Linux (AGL) / Genivi

https://www.automotivelinux.org/

http://genivi.org/

Renesas

http://www.renesas.com/applications/automotive/index.jsp

http://elinux.org/R-Car/Boards/Porter

Intel

http://www.intel.com/content/www/us/en/internet-of-things/overview.html

http://www.theverge.com/2015/1/16/7555647/the-internet-of-things-is-already-a-2-billion-business-for-intel

Samsung

http://pro.clubic.com/actualite-e-business/investissement/actualite-770482-sigfox-samsung.html

https://www.artik.io/

Consortiums IoT

OCF (Open Connectivity Foundation) : https://openconnectivity.org/

IIC (Industrial Internet Consortium) : http://www.iiconsortium.org/

Allseen Alliance :https://allseenalliance.org/