Protéger les systèmes embarqués grâce aux signatures et aux certificats numériques

Une architecture de sécurité doit englober non seulement un dispositif embarqué cible mais aussi tous les points d'accès et les utilisateurs au sein de l'environnement d'exploitation complet. Etant donné qu'aucun réseau n'est sécurisé dans le monde de l'Internet des objets, tous les nœuds distants doivent être authentifiés avant que commandes et données puissent être jugées dignes de confiance. Green Hills Software explique comment les certificats et les signatures numériques permettent une communication sécurisée à travers une authentification mutuelle.

AUTEUR

Darryl Parisien, directeur des ventes, Integrity Secure Services, une société de Green Hills Software.

Lorsqu'il s'agit d'intégrer la sécurité dans une conception, l'environnement au sein duquel ladite conception va évoluer doit déterminer le degré de robustesse exigé. Une architecture de sécurité doit dès lors englober non seulement l'appareil ciblé, mais tous les équipements distants et les utilisateurs appartenant à l'ensemble du système. Même s'il existe des numéros de série, des adresses MAC et des listes blanches et noires, ces dispositifs ne sont pas infaillibles. La plupart des attaques sur des systèmes embarqués se fondent en effet sur la surveillance du trafic réseau pour effectuer de la rétro-ingénierie sur les commandes pour les ré-exécuter ailleurs, dans leur version d'origine ou modifiée.

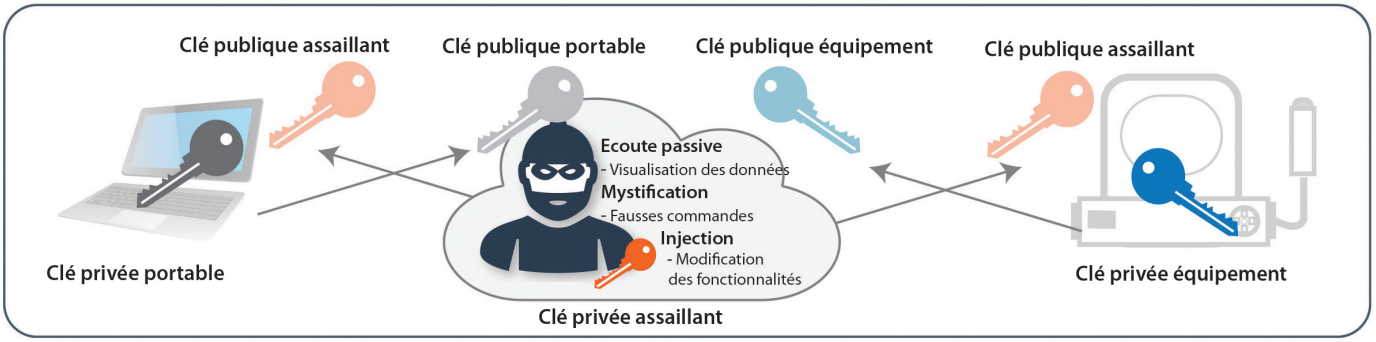
Dans l'univers de l'Internet des objets (IoT), toute entité est inconnue jusqu'à preuve du contraire. Même les réseaux privés sont exposés à des atteintes aux données et sont tout aussi vulnérables que l'Internet. Aucun réseau n'étant sécurisé, tous les points d'extrémité distants doivent être authentifiés avant de pouvoir considérer les commandes et les données associées comme dignes de confiance. Si les êtres humains utilisent des mots de passe, les ordinateurs utilisent, pour leur part, des clés. Si des dispositifs utilisant des clés secrètes sont réalisables dans des environnements limités, la complexité et les coûts nécessaires pour protéger ces clés augmentent proportionnellement au nombre d'appareils connectés. En effet, si une clé secrète est exposée du fait de l'atteinte d'un seul dispositif, tous les autres systèmes présents dans l'environnement seront vulnérables et dans l'incapacité de distinguer un système inconnu d'un système digne de confiance.

Partager des clés secrètes

Les noms d'utilisateur et les mots de passe permettent de produire dynamiquement des clés secrètes uniques, qui sont saisies à la demande et qui n'ont pas besoin d'être stockées dans des mémoires non volatiles, réduisant ainsi le risque de compromission. Cette méthode fonctionne dans certains environnements, mais la plupart des produits n'incorporent pas ce concept d'utilisateur et doivent fonctionner de manière sécurisée dès leur mise sous tension. Ces environnements nécessitent de partager des clés secrètes entre deux équipements pour assurer l'authentification et le chiffrement. Pour autant, comment les clés sont-elles introduites dans le

1 PRINCIPE D'UNE ATTAQUE DU TYPE «MAN-IN-THE-MIDDLE»

Comment un appareil sait-il qu'il communique de manière sécurisée avec le bon équipement ? Les certificats servent à éviter les attaques par intercepteur (man-in-the-middle) au cours de l'échange de clés publiques et à prouver l'identité d'un équipement distant.



système ? Dans les environnements militaires, il existe des ordinateurs portatifs spéciaux (appelés chargeurs de clés), utilisés pour transférer physiquement les clés secrètes dans un système. Ces systèmes possèdent en fait un port de chargement dont la fonction est de recevoir les clés, qui ne sont ainsi jamais exposées. Les chargeurs et les systèmes de gestion de clés les protègent pendant leur transfert jusqu'à ce qu'elles soient stockées de manière sûre dans le périmètre de sécurité d'un appareil.

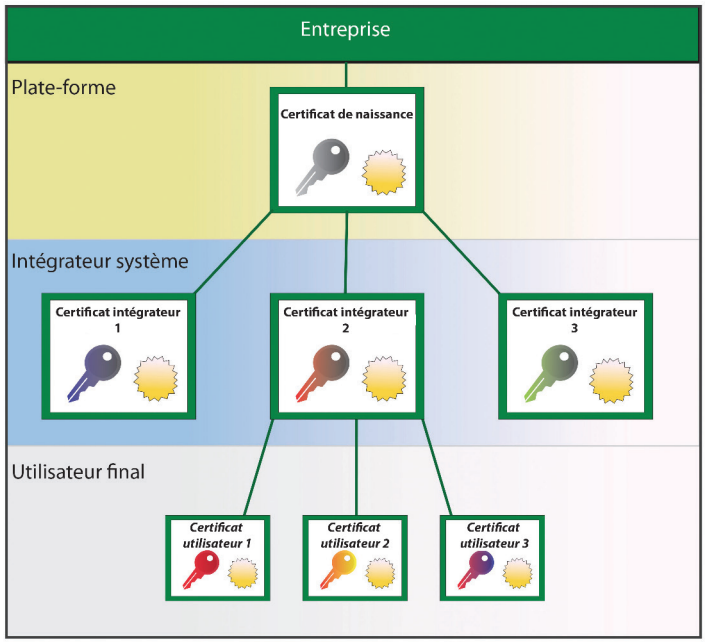
Le chiffrement par clés publiques simplifie la complexité des clés partagées en attribuant à chaque point d'extrémité une paire unique de clés asymétriques utilisée pour le chiffrement et l'authentification. Les bonnes pratiques de sécurité n'utilisent jamais la même clé pour ces deux opérations (chiffrement et authentification). Si un équipement est compromis et qu'une clé privée est exposée, la vulnérabilité sera limitée à ce seul système.

La cryptographie asymétrique permet aux équipements de communiquer de manière sécurisée sans nécessiter un partage préalable des clés. Cependant, comment un appareil sait-il qu'il communique de manière sécurisée avec le bon équipement ? Les certificats servent à éviter les attaques par intercepteur (man-in-the-middle) au cours de l'échange de clés publiques et à prouver l'identité d'un équipement distant (figure 1). Sous sa forme la plus élémentaire, un certificat est constitué de métadonnées indiquant le nom, le numéro de série, les données de durée de validité, etc. associées à la clé publique par une signature numérique émanant d'une Autorité de certification (CA). Grâce à l'authentification mutuelle, les appareils ne tentent jamais de traiter des commandes entrantes sauf celles provenant d'une source valide. Les services IoT d'entreprise sont également protégés car eux aussi ne répondent qu'à des dispositifs dûment identifiés.

Lorsque deux systèmes échangent des certificats, il est possible d'authentifier les clés publiques si elles possèdent l'une et l'autre une signature numérique issue de la même CA. Les deux systèmes accordant leur confiance à cette CA, la confiance peut être étendue au système distant, en supposant toutefois que les clés privées n'aient pas été compromises. Une fois le certificat authentifié et considéré comme digne de confiance, le logiciel peut le lire pour déterminer l'action à effectuer ou les fonctions système à activer. Il est ainsi possible, par exemple, de déclencher le mode de débogage sur des appareils pour un temps limité selon le niveau de formation du technicien.

2 PRINCIPE DE LA GÉNÉRATION ET DE LA GESTION DE CERTIFICATS DANS LES SYSTÈMES EMBARQUÉS

Selon la spécification 802.11ar, le « certificat de naissance » (Birth Certificate) permet d'identifier le fabricant du système, de la carte ou du composant. Il s'agit généralement du premier certificat incorporé dans le système et il est généré lors de sa fabrication. La finalité des certificats « dérivés » dépend du contexte. Il est par exemple possible de vendre une carte électronique à différents intégrateurs, qui souhaitent disposer de systèmes cryptographiques distincts de leurs concurrents.



Générer et gérer des certificats

Il existe différents dispositifs de génération et de gestion de certificats pour aider les développeurs de systèmes embarqués à incorporer des identités numériques dans leurs produits. Ainsi, la spécification IEEE 802.1ar est de plus en plus utilisée pour les produits embarqués, comme par exemple les systèmes de communication en réseau et de supervision industrielle. La spécification 802.1ar répond aux problématiques de gestion et d'utilisation d'un certificat iDevID (Initial Device Identifier) unique et de plusieurs certificats LDevID (Locally Significant Device Identifier).

Le certificat iDevID, appelé aussi « certificat de naissance », permet d'identifier le fabricant du système, de la carte ou du composant. Il s'agit généralement du premier certificat incorporé dans le système et il est généré lors de sa fabrication. En revanche, la finalité des certificats LDevID dépend du contexte. Par exemple, il est possible, pour un produit donné, de créer un certificat LDevID dans le but de protéger les profils et les données d'un client. Autre exemple, il est possible de vendre une carte électronique à différents intégrateurs, qui souhaitent disposer de systèmes cryptographiques distincts de leurs concurrents. En effet, si le fabricant de la carte souhaite transmettre des mises à jour de logiciels sécurisées et collecter des contenus à distance, les intégrateurs souhaitent eux que leurs droits de propriété intellectuelle soient protégés. Ainsi, il est possible de générer un certificat LDevID pour chaque intégrateur pour lui permettre de communiquer de manière sécurisée à l'intérieur de son propre périmètre local, et de protéger ses données de tout risque d'atteinte (figure 2).

Pour incorporer des certificats dans un dispositif embarqué, les développeurs et les fabricants doivent prendre en compte les opérations suivantes :

- **Programmation du certificat racine de l'autorité de certification (Root CA) dans une mémoire immuable**

Cette opération permet d'empêcher un attaquant de remplacer ce certificat par autre chose.

- **Génération de paires de clés asymétriques**

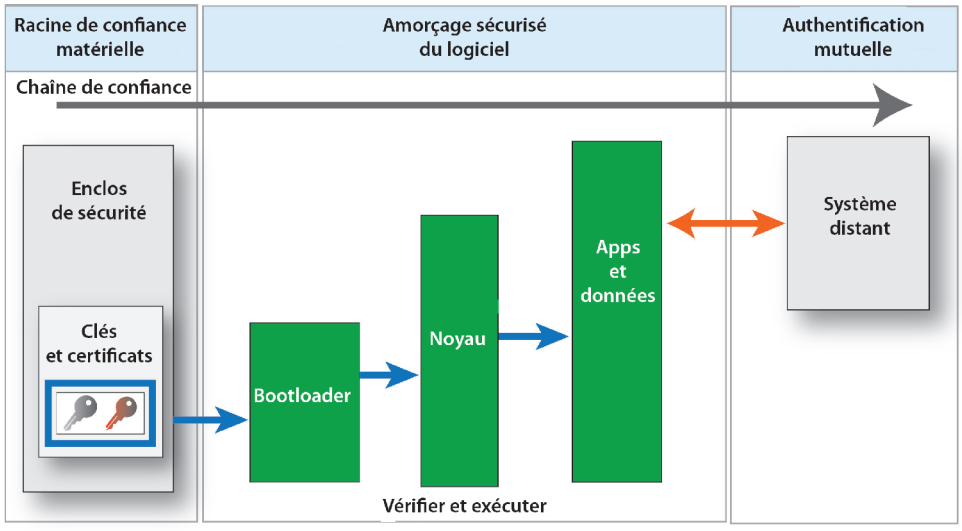
Les clés privées ne doivent jamais être exposées en dehors du dispositif, ce qui suppose qu'il soit possible de générer des nombres aléatoires appropriés.

- **Protection des clés privées**

Les attaques par usurpation sont possibles si les clés privées sont compromises.

2 PRINCIPE DE LA VÉRIFICATION DE L'AUTHENTICITÉ D'UN LOGICIEL

Lors de chaque mise sous tension, l'amorçage sécurisé vérifie l'authenticité de chaque couche du logiciel avant d'autoriser son exécution. Cette opération permet de s'assurer que le logiciel n'est pas altéré et provient bien d'une source valide. En effet, un composant n'est jamais exécuté tant qu'il n'a pas été dûment authentifié.



- **Envoi de demandes de signature de certificats au système de l'Autorité de certification**

Si vous utilisez un système d'autorité de certification hébergé, quel est l'impact sur la production en cas de panne de réseau ?

- **Réception des certificats et protection du stockage**

De quelle manière les certificats et les clés sont-ils protégés sur l'appareil ?

- **Chargement des listes de révocation initiales**

Cette opération concerne l'ensemble du processus de gestion des stocks et le suivi des systèmes et des RMA accrédités.

Générer des certificats sur les lignes de production

Un nombre croissant de développeurs de produits créent leurs propres infrastructures de clés publiques (PKI) et génèrent des certificats directement sur les lignes de production. Les clés privées restent ainsi protégées à l'intérieur de chaque appareil et la production est insensible aux arrêts de fonctionnement d'Internet. Le système DLM (Device Lifecycle Management) d'integrity Security Services protège les clés CA d'une entreprise contre toute atteinte à la sécurité des données possible depuis les réseaux informatiques sur des sites de fabrication distants ou tiers. Cette approche permet une très grande disponibilité des opérations de génération de certificats. Le système DLM répond aux besoins des développeurs qui souhaitent incorporer des certificats adaptés à leurs chaînes de conception et de logistique, et ce, sans avoir à construire et maintenir leur propre infrastructure de clés publiques.

L'une des questions les plus importantes concernant l'authentification est de savoir comment accorder de la confiance aux autres si ne sommes pas en mesure de nous accorder de la confiance à nous-mêmes. Malheureusement, il ne s'agit pas d'une question philosophique. Si le logiciel d'un système est compromis, c'est toute la chaîne de confiance qui est rompue depuis l'origine, et il n'y a plus aucune garantie. Un logiciel piraté peut contourner les vérifications, accepter n'importe quel certificat et modifier le contenu des messages. En outre, insuffisamment protégées, les clés peuvent être compromises et utilisées pour attaquer d'autres dispositifs en manipulant les commandes et les données. À cela s'ajoutent les accès dérobés (backdoors) qui, une fois ouverts, permettent de collecter et d'envoyer des données, rendant alors inopérants les dispositifs de sécurité et les schémas d'authentification.

Avant qu'un système embarqué puisse se faire suffisamment confiance pour authentifier des équipements distants, il doit vérifier que son logiciel n'a pas été modifié. Il existe pour cela le processus d'« amorçage sécurisé », qui prévoit la vérification du logiciel système avant exécution (figure 3). Lors de chaque mise sous tension, l'amorçage sécurisé vérifie l'authenticité de chaque couche du logiciel avant d'autoriser son exécution. Cette opération permet de s'assurer que le logiciel n'est pas altéré et provient bien d'une source valide. En effet, un composant n'est jamais exécuté tant qu'il n'a pas été dûment authentifié.

Vérifier l'authenticité du logiciel

Le hachage et les sommes de contrôle permettent de vérifier l'intégrité du logiciel, mais pas son authenticité. Si un attaquant modifie le hachage simultanément au code, un logiciel malveillant peut être exécuté en passant inaperçu. L'authentification est effectuée en ajoutant une signature numérique au hachage grâce à une clé asymétrique. L'infrastructure de sécurité de l'entreprise protège la clé privée et associe une signature numérique au logiciel diffusé. La clé publique correspondante est programmée dans le dispositif en fabrication, puis utilisée au cours de la vérification. À ce stade, si un attaquant modifie à la fois le code et le hachage, il ne peut pas mettre à jour la signature numérique sans la clé privée correspondante pour la régénérer. Integrity Security Services intervient auprès d'entreprises opérant dans tous les secteurs d'activité pour déployer des solutions d'amorçage sécurisé, avec des infrastructures de signatures numériques assurant la protection des clés privées pour éviter toute exposition et l'intégration dans les processus traditionnels de construction de logiciels.

Aujourd'hui, l'architecture de sécurité des produits IoT connectés en réseau doit prendre en compte davantage d'inconnues qu'auparavant. Chaque dispositif ajouté au réseau apporte son lot d'inconnues (menaces et risques de conflits), et les coûts d'assistance et de développement qui les accompagnent. Limiter ces inconnues suppose de contrôler la communication par un processus d'authentification pour les seuls équipements dignes de confiance. La chaîne de la confiance sera ainsi protégée en commençant par contrôler le matériel, puis vérifier le logiciel avant d'étendre le processus aux appareils distants. ■