RFC 9124 : A Manifest Information Model for Firmware Updates in Internet of Things (IoT) Devices

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 15 janvier 2022

Date de publication du RFC : Janvier 2022

https://www.bortzmeyer.org/9124.html

On le sait, la sécurité des gadgets nommés « objets connectés » est abyssalement basse. Une de raisons (mais pas la seule, loin de là !) est l'absence d'un mécanisme de mise à jour du logiciel, mécanisme qui pourrait permettre de corriger les inévitables failles de sécurité. Le problème de la mise à jour de ces machins, souvent contraints en ressources, est très vaste et complexe. Le groupe de travail SUIT <https://datatracker.ietf.org/wg/suit/> de l'IETF se focalise sur **un** point bien particulier : un format de manifeste permettant de décrire une mise à jour. Ce RFC décrit le modèle de données des informations qui seront placées dans ce manifeste.

Donc, que faut-il indiquer pour permettre une mise à jour du logiciel (et des réglages) d'un objet connecté ? Le groupe de travail <https://datatracker.ietf.org/wg/suit/> est parti des expériences concrètes, de scénarios et de menaces, pour arriver à la liste que contient ce RFC. Certes, cette liste n'est pas exhaustive (cela serait impossible) mais donne quand même un bon point de départ. Attention, ce RFC décrit le modèle d'information (cf. RFC 3444 [1]), pas le format du manifeste (qui sera en CBOR, et fera l'objet d'un futur RFC).

Pour suivre le contenu de ce RFC, il vaut mieux avoir déjà lu le premier RFC du groupe de travail, le RFC 9019, qui explique l'architecture générale du système, ainsi que le RFC 8240, qui était le compte-rendu d'un atelier de réflexion sur cette histoire de mise à jour des objets connectés..

La (longue) section 3 du RFC est consacrée à énumérer tous les éléments à mettre dans le manifeste. Je ne vais pas reproduire toute la liste, juste quelques éléments intéressants. Notez que pour chaque élément, le RFC précise s'il **doit** être présent dans le manifeste ou bien s'il est facultatif. Le premier

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc3444.txt

élément listé est l'obligatoire identificateur de version de la structure du manifeste, qui permettra de savoir à quel modèle ce manifeste se réfère.

Autre élément obligatoire, un numéro de séquence, croissant de façon monotone, et qui permettra d'éviter les reculs accidentels (« mise à jour » avec une version plus ancienne que celle installée). On peut utiliser pour cela une estampille temporelle (si on a une horloge sûre).

Ensuite, le RFC recommande (mais, contrairement aux deux précédents éléments, ce n'est pas obligatoire) de placer dans le manifeste un identificateur du fournisseur de la mise à jour. Il n'est pas prévu pour d'autres comparaisons que la simple égalité. Le format recommandé pour cet identificateur est un UUID (RFC 9562) de « version 5 », c'est-à-dire formé à partir d'un nom de domaine, qui est forcément unique. L'avantage des UUID (surtout par rapport au texte libre) est leur taille fixe, qui simplifie analyse et comparaison. Si on veut un identificateur de fournisseur qui soit lisible par des humains, il faut le placer dans un autre élément.

L'identificateur de la classe ("*class ID*"), indique un type de machines, les machines d'une même classe acceptant le même logiciel (cette acceptation dépend du matériel mais aussi d'autres facteurs comme la version du microcode). Il doit être unique par identificateur de fournisseur (et, en cas de vente en marque blanche, il doit être fourni par le vrai fournisseur, pas par le vendeur). Là aussi, un UUID de « version 5 » est recommandé. Il ne doit pas dépendre juste du nom commercial, un même nom commercial pouvant recouvrir des produits qui n'acceptent pas les mêmes mises à jour. Si un objet peut recevoir des mises à jour indépendantes, pour différents composants de l'objet, il faut des identificateurs de classe différents (surtout si certains objets du fournisseur utilisent une partie des mêmes composants, mais pas tous ; l'identificateur doit identifier le composant, pas l'objet).

Le manifeste contient aussi (mais ce n'est pas obligatoire), une date d'expiration, indiquant à partir de quand il cesse d'être valable.

Le manifeste peut (mais ce n'est pas obligatoire) contenir directement l'image utilisée pour la mise à jour du logiciel de l'objet. Cela peut être utile pour les images de petite taille ; plus besoin d'une étape supplémentaire de téléchargement.

Par contre, l'indication du format de l'image, qu'elle soit directement incluse ou non, est obligatoire, ainsi que celle de la taille de la dite image.

Question sécurité, le RFC impose également la présence d'une signature du manifeste. Le manifeste peut aussi contenir des éléments qui vont servir à établir si on a une délégation sûre depuis une autorité reconnue : des "*Web Tokens*" en CBOR (RFC 8392), avec peut-être des preuves du RFC 8747.

La sécurité est au cœur des problèmes que traite ce RFC. Après cette liste d'élements facultatifs ou obligatoires dans un manifeste, la section 4 de notre RFC expose le modèle des menaces auxquelles il s'agit de faire face. Le RFC rappelle que la mise à jour elle-même peut être une menace : après tout, mettre à jour du logiciel, c'est exécuter du code distant. Juste répéter « il faut mettre à jour le logiciel de sa brosse à dents connectée » ne suffit pas, si le mécanisme de mise à jour permet d'insérer du code malveillant. (Et encore, le RFC est gentil, tendance bisounours, il ne rappelle pas que l'attaquant peut être le fournisseur, envoyant du code nouveau pour désactiver certaines fonctions ou pour espionner l'utilisateur, voir les exemples de Hewlett-Packard <https://boingboing.net/2016/09/19/hp-detonates-its-timebomb-pri.html> et de Sony. Sans même parler de la possibilité d'une attaque contre la chaine de développement comme celle contre SolarWinds.)

L'analyse part du modèle STRIDE <https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx>. Je ne vais pas citer toutes les menaces possibles (il y en a beaucoup !), lisez le RFC pour avoir une vue complète. Notez que les attaques physiques contre les objets (ouvrir la boite et bricoler à l'intérieur) ne sont pas incluses.

Évidemment, la première menace est celle d'une mise à jour qui serait modifiée par un attaquant. Le code correspondant serait exécuté, avec les conséquences qu'on imagine. La signature prévient cette attaque.

Ensuite, le cas d'une vieille mise à jour, qui était honnête et signée, mais n'est plus d'actualité. Si elle a une faiblesse connue, un attaquant pourrait essayer de faire réaliser une « mise à jour » vers cette vieille version. Le numéro de séquence dans le manifeste, qui est strictement croissant, est là pour protéger de cette attaque.

Autre risque, celui d'une mise à jour signée mais qui concerne en fait un autre type d'appareil. Appliquer cette mise à jour pourrait mener à un déni de service. La protection contre ce risque est assurée par l'identificateur du type d'objet.

Le RFC liste la menace d'une rétro-ingénierie de l'image. Outre que cela ne s'applique qu'au logiciel privateur, du point de vue de la sécurité, cela n'est pas crucial, puisqu'on ne compte pas sur la STO. Si on y tient quand même, le chiffrement de l'image (qui n'est pas obligatoire) pare ce risque.

Sinon la section 4.4 contient des scénarios typiques d'utilisation, où une histoire décrit les acteurs, les menaces, les solutions possibles, rendant ainsi plus vivants et plus concrets les problèmes de sécurité étudiés dans ce RFC.