Que signifie le Cyber Resilience Act pour les fabricants de produits connectés ?

La loi sur la cyber-résilience ou Cyber Resilience Act (CRA) de l'Union européenne est une nouvelle législation proposée par la Commission européenne visant à améliorer la posture de cybersécurité des organisations utilisant des actifs numériques. Elle vise à garantir que les entreprises disposent des politiques, procédures et technologies nécessaires pour se protéger, ainsi que leurs clients, contre les menaces cybernétiques. Revue de détail par Marc Le Guyader, de la société eShard.

AUTEUR

Marc Le Guyader, Sales and Business Development Director, eShard.

Faisant suite à l'augmentation du nombre et du coût des cyberattaques réussies, qui ont été estimées à entraîner des coûts mondiaux annuels de 5,5 billions d'euros en 2021, la législation du Cyber Resilience Act (CRA) vise à établir des réglementations communes en matière de cybersécurité pour les fabricants et les développeurs de produits comportant des éléments numériques sur le marché de l'Union européenne. Cette loi est unique en son genre et témoigne de l'analyse de la Commission européenne selon laquelle les risques cybernétiques sont une question d'importance sociale, politique et économique. Actuellement, il n'existe pas de norme cohérente en matière de cybersécurité pour tous les produits exploitant des actifs numériques sur le marché de l'UE, et le CRA vise à remédier à cette situation.

Pour rappel, ce texte concerne tout produit connecté déployé sur le terrain, que ce soit pour le grand public ou le monde professionnel. Selon leur nature, les fabricants de produits devront soit procéder à une auto-évaluation de leur produit (catégorie par défaut), soit effectuer des tests par eux-mêmes (Classe I critique), soit encore passer par un laboratoire de certification (Classe II critique) (voir figure 1).

Que devront faire les OEM pour se préparer au CRA ?

Pour répondre à cette question, la première chose à faire est de considérer la notion de défense en profondeur (defense in-depth). Pour ceux qui ne sont pas familiers avec ce concept, la defense in-depth signifie que la sécurité est mise en œuvre à chaque couche du produit (figure 2). Une couche peut être le matériel (par exemple, une puce intégrant des actifs cryptographiques), le logiciel bas niveau (par exemple, un bootloader), le système d'exploitation (Linux), le logiciel ou les bibliothèques applicatives (par exemple, une pile de protocoles réseau, une bibliothèque de cryptographie implantée en logiciel) ou encore le service (par exemple, la connexion à un serveur back-end).

Pour partir sur des bases solides, les fabricants de produits électroniques devront de préférence sélectionner du matériel et des logiciels conformes individuellement aux règles du CRA. Cependant cela ne sera pas suffisant, car en intégrant et en modifiant éventuellement différents composants matériels et logiciels dans leur produit, ainsi que leur propre code, ils pourraient par erreur ou par ignorance - par exemple, en l'absence de directives de mise en œuvre fournies par le fournisseur en matière de sécurité - introduire des vulnérabilités dans le système global. Ce qui pourrait se produire lors du lancement du produit mais également lors d'une mise à jour du logiciel.

Une conséquence immédiate découle de cette approche : les fabricants de produits devront mettre en place une politique de sécurité ou la renforcer afin de transformer leur cycle de développement de produits en un cycle de développement de produits sécurisé, ce qui impliquera principalement :

- d'identifier et de suivre les actifs stockés et manipulés par le produit.
- d'appliquer les lignes directrices de sécurité fournies par leurs fournisseurs, le cas échéant, lors de l'intégration de leurs composants dans le produit.
- de surveiller les vulnérabilités et expositions communes – CVE, Common Vulnerabilities and Exposures (*) – du matériel et des logiciels des composants qu'ils intègrent.
- de tester les fonctions critiques – du

ESHARD, POUR UNE ANALYSE PROFONDE DE LA CYBERSÉCURITÉ

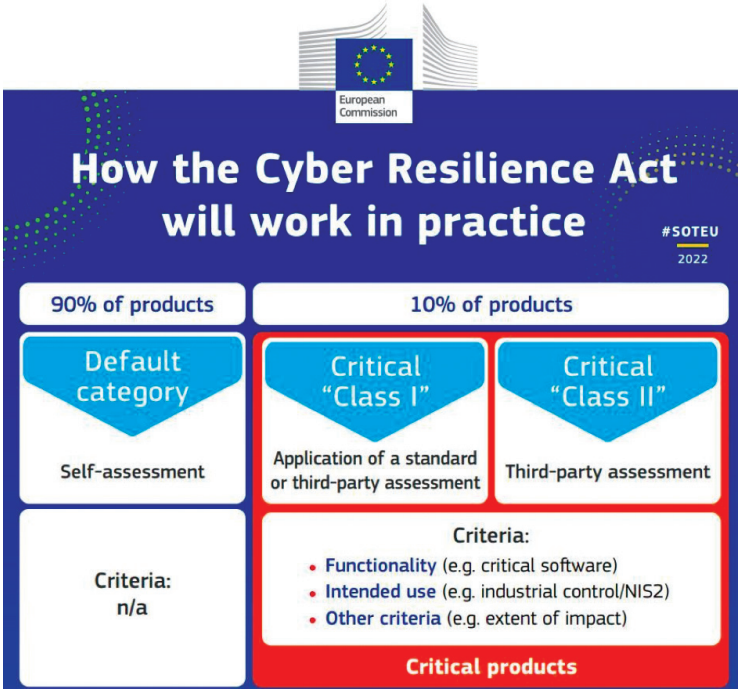
■ eShard est une jeune entreprise française créée en 2015, basée à Pessac (à côté de Bordeaux) et spécialiste de la sécurité des systèmes embarqués utilisés dans des environnements de pointe comme le logiciel des puces électroniques dans des systèmes de défense, des environnements grand public comme les applications mobiles ou dans des objets communicants du quotidien en charge du stockage et de l'échange de nos données personnelles.

■ La société, forte d'environ 35 employés, délivre aux concepteurs ou utilisateurs d'objets connectés, les moyens de maîtriser les risques de cybersécurité et de s'assurer que ces objets intègrent le bon niveau de protection en comprenant la menace, en menant des vérifications automatiques et en maintenant la connaissance des attaques.

■ Le savoir-faire d'eShard s'incarne dans des plateformes SaaS, des logiciels, des formations et transferts de compétence techniques et des services de tests de sécurité.

1 COMMENT LE CYBER RESILIENCE ACT FONCTIONNE EN PRATIQUE

Selon leur nature, les fabricants de produits devront soit procéder à une auto-évaluation de leur produit - catégorie par défaut - soit effectuer des tests par eux-mêmes - Classe I critique - soit encore passer par un laboratoire de certification - Classe II critique.



point de vue de la sécurité – manipulant ou donnant accès aux actifs du produit.

Un exemple de test de sécurité du schéma de certification Arm PSA

Prenons un exemple tiré du schéma de certification PSA (Platform Security Architecture) issu des travaux de la société Arm, et désormais bien connu de l'industrie de l'IoT : le cas de test de la version 1.2 des méthodes d'attaque certifiées par PSA, appelé « Glitch against initialization » (défaillance lors de l'initialisation).

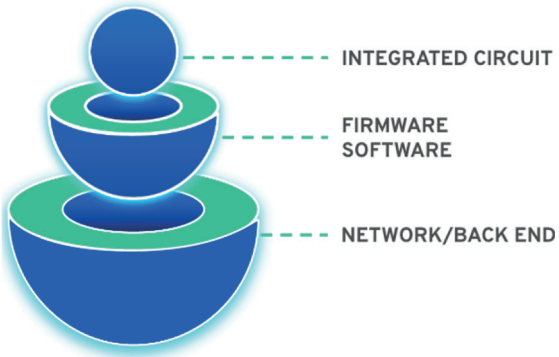
Ce test a pour objectif de vérifier que le produit testé n'est pas sensible à une perturbation appliquée à une entrée/sortie de la puce pendant le processus de démarrage et d'initialisation du produit. Cette technique peut être utilisée par des attaquants pour perturber cette phase critique et obtenir un comportement inattendu tel que la suppression de la vérification de la signature du logiciel et l'exécution d'un logiciel malveillant donnant à l'attaquant des privilèges de contrôle sur le produit.

Comment dès lors le fabricant du produit peut-il s'assurer que ce test « Glitch against initialization » est réussi sur son produit ? Nous pouvons raisonnablement penser, et certainement vérifier, que ce test a été effectué et réussi par le fabricant de la puce. Mais que se passe-t-il si des modifications apportées au bootloader ou à son environnement modifient inopinément le comportement du produit et introduisent une vulnérabilité ? La seule façon de le découvrir est de le tester. Or tester les protections dans un produit est une tâche difficile car la plupart des protections n'ont aucun impact fonctionnel. Elles agissent en arrière-plan en cas d'événement malveillant. On pourrait dire qu'elles sont «transparentes» et nécessitent un environnement de test spécifique simulant les attaques pour déclencher les protections et exécuter le code correspondant. C'est le cas de notre scénario de défaillance. De plus, la plupart des fabricants de produits ne disposent pas d'une configuration de test pour réaliser un tel test et ils n'en ont peut-être même jamais entendu parler.

Quoi qu'il en soit, la bonne nouvelle est que les développeurs n'ont pas besoin de tout cet arsenal de test, car ils n'ont pas pour objectif de tester la puce sur laquelle le logiciel est exécuté – cela avait été fait par le fabricant de la puce – mais uniquement de tester si les modifications qu'ils ont apportées au logiciel fourni avec la puce n'ont pas induit de régression dans ce test de défaillance.

1 L'APPROCHE «DEFENSE IN-DEPTH»

La defense in-depth ou défense en profondeur signifie que la sécurité est mise en œuvre à chaque couche du produit, à savoir le matériel, par exemple une puce intégrant des actifs cryptographiques, le logiciel bas niveau, par exemple un bootloader, le système d'exploitation, le logiciel ou les bibliothèques applicatives, par exemple une pile de protocoles réseau, ou encore le service, par exemple la connexion à un serveur back-end.



Vous développez du logiciel, travaillez dans un environnement d'émulation

Un outil de test tel que esFirmware(**) mis au point par la société française eShard fournit un environnement d'émulation dans lequel il est possible d'exécuter tout ou partie d'un logiciel, et intègre des connaissances sur le test « Glitch against initialization ». Associé à un cas d'utilisation pratique, il autorise le fabricant à tester, à minimiser les risques et à instaurer un bon niveau de confiance sans avoir l'obligation d'investir dans le développement d'une nouvelle expertise.

Tout ce qui vient d'être décrit sur ce cas de test de défaillance n'est qu'un exemple pour illustrer ces enjeux. Mais cette approche peut être étendue à de nombreuses autres attaques qui doivent être testées sur le produit final mis en production afin d'éviter l'introduction de toute régression de sécurité au cours du cycle de développement.

In fine, que devront faire les OEM pour se préparer au CRA ? D'abord identifier les menaces auxquelles les dispositifs connectés sont exposés, puis comprendre les vecteurs d'attaque et enfin effectuer des tests de non-régression de sécurité. ■

(*) Common Vulnerabilities and Exposures ou CVE est un dictionnaire des informations publiques relatives aux vulnérabilités de sécurité. Le dictionnaire est maintenu par l'organisme Mitre, soutenu par le département de la Sécurité intérieure des États-Unis.

(**) Le framework esFirmware d'eShard offre la possibilité d'analyser en profondeur un firmware embarqué dans un circuit contre les attaques physiques en exploitant un moteur d'émulation qui interfère dans l'exécution du runtime pour simuler des observations ou des perturbations proches de la réalité physique.