Attaque contre un HSM: « Unwrapping the chrysalis »

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 24 juin 2010

https://www.bortzmeyer.org/attack-hsm.html

Peut-on attaquer un HSM au point de récupérer la clé privée, qui ne devrait jamais sortir ? Oui, montre cet article, qui a réussi contre un modèle de HSM particulièrement mal fait du point de vue matériel (mais plus sérieux sur le logiciel).

Les HSM sont devenus des composants indispensables des systèmes de sécurité basés sur la cryptographie. Par exemple, la grande majorité des registres de noms de domaines qui signent leur zone avec DNSSEC utilisent un HSM.

Ces engins spécialisés génèrent des couples clé privée / clé publique et gardent la clé privée à l'intérieur, sans aucun moyen de la sortir sans qu'elle soit chiffrée. Des protections matérielles garantissent que même un méchant qui a un accès physique au HSM ne pourra l'ouvrir ou que, s'il le fera, la clé privée sera automatiquement effacée (si le HSM est conforme à la norme « FIPS 140-2 level 4 »). Comme la clé privée ne sort pas, toutes les opérations cryptographiques (signature, par exemple) doivent être faites dans le HSM, qui dispose donc d'un processeur pour cela.

En matière de sécurité, il n'y a pas de solution parfaite, pas de système invulnérable, pas de bouclier sans épée capable de le percer. Les HSM n'echappent pas à cette règle et l'article « "*Unwrapping the Chrysalis*" <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-592.pdf> » explique une attaque réussie.

J'ai été plutôt déçu par la partie matérielle de cet article, où j'espérais apprendre des méthodes de très haute technologie, genre roman de Michael Crichton : le HSM en question, un Luna Chrysalis, a été percé en un rien de temps, malgré sa certification FIPS 140-2 level 3. L'article date de 2004, j'espère que les HSM ont progressé depuis et que les certificateurs font leur travail plus sérieusement !

La partie logicielle, quoique pas spécifique aux HSM (c'est de la rétro-ingénierie de code ARM classique) est la partie la plus longue de l'article, et celle qui a causé le plus de difficultés aux auteurs. Pourtant, le logiciel n'utilisait apparemment aucune des techniques de dissimulation dont se servent, par exemple, les auteurs de virus. Mais il était très complexe, d'autant plus qu'il devait gérer beaucoup

de cas. Les auteurs ont donc dû entamer une longue et douloureuse route, notant soigneusement tout ce qu'ils découvraient, pour établir peu à peu une carte du logiciel.

N'étant pas familier avec les techniques de rétro-ingéniérie, j'ai appris des choses, comme l'importance de **nommer** tous les objets identifiés. Par exemple, chaque fonction repérée dans le code reçoit un nom, même sans sémantique (comme « Fred », « George » ou « Juliet »). Cela facilite grandement les communications au sein de l'équipe.

Ils ont fini par reconstituer le protocole qui permet à deux de ces HSM de synchroniser leurs clés et cela a permis de finir par sortir une clé privée (avec la complicité d'un "*Security Officer*", toutefois). Les HSM ne sont donc pas invincibles.

---

https://www.bortzmeyer.org/attack-hsm.html