OpenSSH passe à l'authentification FIDO

Vous avez des dispositifs à la norme FIDO U2F ? OpenSSH les prend désormais en charge pour renforcer les procédures d'authentification.

Ils peuvent être associés à deux types de clés exploitant la cryptographie sur courbes elliptiques (ECDSA P-256 et Ed25519).

OpenSSH inclut un *middleware* qui gère la communication avec les *tokens* FIDO sur USB. Activé par défaut sur OpenBSD, il requiert libfido2.

Les clés générées se composent de deux éléments. L'un est stocké sur disque. L'autre réside dans le *token*, auquel il est spécifique.
La combinaison de ces deux éléments sert à valider l'authentification.

Pour les *tokens* utilisés sur plusieurs machines, gérer la partie stockée sur disque peut se révéler contraignant.
La norme FIDO2 contourne cette difficulté en permettant la récupération de cette partie à partir du *token*. Il est toutefois conseillé de sécuriser cette opération avec un PIN.

Service minimum ?

« *C'est la moindre des choses à faire si on veut protéger sérieusement ses comptes* », disait dernièrement Microsoft à propos de l'authentification multifacteur.

Le groupe américain estime que les clients qui activent ce mécanisme ont « 99,9 % de chances en moins » de voir leurs comptes piratés.

Google est un peu plus précis : ajouter un numéro de téléphone à son compte permet de bloquer la majorité des attaques de phishing non ciblées (96 % si le deuxième facteur est code envoyé un SMS ; 99 % s'il s'agit d'une notification à valider).

Parallèlement à la prise en charge de FIDO U2F, OpenSSH retire SHA-1 de la liste des algorithmes acceptés pour signer des certificats. La désactivation de l'algorithme de signature ssh-rsa (qui dépend de SHA-1) est sur la feuille de route.

Illustration principale via *Shutterstock.com*