Dirty Pipe : pourquoi cette faille Linux est critique

Vous souvenez-vous de Dirty Cow ? Fin 2016, on avait eu connaissance de cette faille présente dans le noyau Linux depuis près de dix ans. À la racine, une situation de concurrence. Plusieurs processus disposaient simultanément d'une même ressource, alors que chacun d'eux pensait en avoir l'usage exclusif. Cela perturbait l'exécution et ouvrait une brèche permettant d'obtenir des droits en écriture sur des zones mémoire normalement en lecture seule.

On en reparle après la découverte d'une vulnérabilité aux conséquences similaires. Elle est exploitable à partir de la version 5.8 du noyau. Son matricule : CVE-2022-0847. Celui qui l'a dénichée l'a appelée Dirty Pipe. Et pour cause : elle repose sur les *pipes*, ces tuyaux monodirectionnels qui permettent la communication entre processus.

La faille vient d'être révélée au public. Mais les premiers signaux ayant mené à sa découverte remontent à avril 2021. Le déclencheur : un ticket de support adressé à CM4all (plate-forme de création de sites web fondée sur WordPress). Un client se plaignait de ne pas pouvoir décompresser des journaux d'accès qu'il avait téléchargés. L'un des fichiers sur le serveur de *logs* était effectivement corrompu. Plus précisément au niveau du CRC (contrôle de redondance cyclique, destiné à détecter les erreurs de transmission ou de transfert).

Une faille introduite en deux temps

Au fil des mois, le problème s'est reproduit. Avec, à chaque fois, un problème au niveau de la valeur CRC. Celle-ci était tout simplement la même sur tous les fichiers corrompus.
De fil en aiguille, celui qui allait découvrir Dirty Pipe en est arrivé à coder deux programmes en C. L'un écrivait en boucle la chaîne de caractères « AAAAA » dans un fichier. L'autre transférait ces données dans un *pipe*, puis écrivait « BBBBB » dans ce même *pipe*. Surprise : « BBBBB » apparaissait finalement dans le fichier, alors que la chaîne n'avait été envoyée qu'au *pipe*, sans permission en écriture.

La faille est apparue en deux temps. D'abord avec un bug introduit dans Linux 4.9 et permettant de créer des références arbitraires à des caches de pages. Puis avec la refonte, dans Linux 5.8, de la gestion des fusions au niveau du tampon des *pipes*. Combinés, les deux éléments permettaient d'écrire des données dans un cache simplement en les injectant dans un *pipe*.

Dirty Pipe, (presque) passe-partout

Les conséquences de Dirty Pipe ne sont pas toujours visibles. Ou du moins, elle sont souvent éphémères. Le cache n'écrit effectivement sur le disque que s'il pense qu'une page a été corrompue (= qu'elle est « sale », *dirty* en anglais). Dans le cas contraire, la compression des *logs* se fait normalement et les traces disparaissent au redémarrage ou lorsque le noyau libère le cache pour récupérer de la mémoire.

Ce caractère éphémère permet des attaques d'autant plus discrètes. Mais pas moins puissantes : les écritures peuvent se faire sans permissions (sinon un accès en lecture), sans contraintes de *timing* et à presque tout emplacement mémoire. Elles fonctionnent autant sur les fichiers censés être immuables que sur les *snapshots* btrfs et les volumes en lecture seule de type CD-ROM.

L'équipe de développement du noyau Linux a été avertie le 20 février dernier. Le lendemain, l'équipe sécurité d'Android recevait elle aussi une alerte. Un correctif est disponible depuis le 23 février dans les versions 5.01.102, 5.15.25 et 5.16.11 du *kernel*.

Photo d'illustration © isak55 – Shutterstock