E. Un peu de lecture : e-a

Plongeon dans les appels systèmes Windows

L'article explique un mécanisme important dans les systèmes Windows, comment un programme peut demander au système d'effectuer une tâche plus sensible.

En informatique, un programme n'a pas tous les droits, ceux qui fonctionnent dans ce qu'on appelle le mode utilisateur (ring 3) sont limités.

Par exemple, ils ne peuvent pas accéder directement au disque dur ou à la mémoire du noyau. Pour ça, ils doivent faire appel à des fonctions du système, qui, elles, tournent en mode noyau (ring 0), c'est-à-dire avec tous les droits.

Pour illustrer cela, l'auteur prend un petit programme en C qui utilise une fonction classique de Windows appelée FindFirstFile. En regardant ce programme dans un débogueur, on découvre que cette fonction ne fait pas tout elle-même. En réalité, elle appelle d'autres fonctions plus profondes, couche après couche, jusqu'à arriver à une fonction très spéciale dans Windows, un appel système.

Avant que cet appel soit lancé, un numéro identifiant la fonction système est placé dans un registre spécial du processeur (EAX).

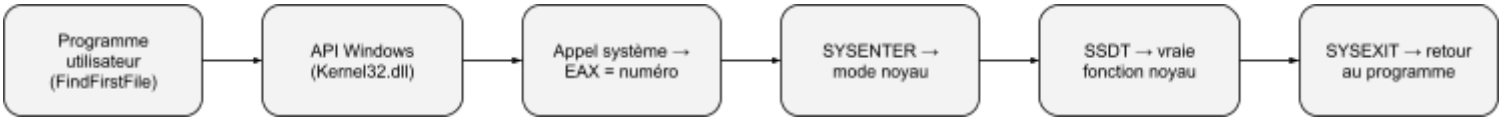
Ensuite, une instruction appelée SYSENTER est exécutée. Cette instruction dit au processeur « Maintenant, passe en mode noyau pour exécuter cette action ». C'est comme si le programme appuyait sur un bouton « autorisation spéciale » pour que le système prenne le relais.

À ce moment-là, le processeur va utiliser trois informations qui ont été placées dans des registres spéciaux appelés MSR (Model Specific Registers). Ces informations lui indiquent dans quel segment de code du noyau il doit aller, à quelle adresse il doit commencer l'exécution, et où se trouve la pile mémoire à utiliser pendant cette phase.

Mais attention, l'adresse d'exécution donnée ne pointe pas directement vers la vraie fonction du système qu'on veut appeler. En réalité, le processeur commence toujours par passer par une routine spéciale du noyau appelée KiFastCallEntry. C'est un peu comme un point d'entrée central qui va rediriger la demande vers la bonne fonction.

Pour savoir précisément quelle fonction doit être exécutée, le système s'appuie sur une table appelée SSDT (System Service Dispatch Table). Cette table contient toutes les fonctions internes que le noyau peut exécuter, chacune étant associée à un numéro. Ce numéro est placé dans le registre EAX juste avant l'appel à SYSENTER. Il sert d'index pour retrouver la bonne fonction dans la table.

Une fois que la fonction du noyau a été exécutée, une autre instruction spéciale, appelée SYSEXIT, est utilisée pour revenir au programme d'origine, et repasser du mode noyau au mode utilisateur, comme si on refermait la porte du système après avoir terminé l'action demandée.



Fonctionnement d'un appel système Windows

Enfin, l'auteur parle d'une partie sécurité, il est possible d'intercepter ou de modifier ces appels pour surveiller ou changer le comportement du système. Cela s'appelle le hooking. On peut le faire du côté utilisateur (en modifiant des pointeurs vers les fonctions), ou directement dans le noyau (ce qui est plus risqué, mais plus puissant).