HTS Exam Project

Baptiste Nehmé

Which

Le script which est assez basique, le principe est de parcourir le PATH de l'utilisateur afin de trouver les paths des binaires. Pour cela, on utlise l'IFS que l'on modifie en « : » pour pouvoir boucler sur la variable « PATH » directement. Il faut bien sur s'assurer de restaurer l'IFS ensuite.

Le seul paramètre accepté par le which d'OpenBSD est « -a » qui ne peut être place qu'en premier argument.

La spécificité majeure du which d'OpenBSD est le code de sorti : 0 si tout est trouvé, 2 si aucun des arguments n'est trouvé, et 1 si une partie des arguments ont été trouvé.

Which gère aussi le cas où la variable « PATH » est vide, en la remplaçant avec le « PATH » par default.

Forbidden syscalls

Pour cet exercice, après avoir installé OpenBSD 7.3, il nous suffit de créer deux fichiers basiques : un main.c et un mygetpid.s.

Le premier est un simple programme C dont le but est d'appeler la fonction « mygetpid » et d'afficher son résultat sur la sortie standard.

Le second fichier est la fonction « mygetpid » définit en assembleur qui est un simple appel au syscall « getpid ».

Si nous compilons ce programme sans options spéciale, l'exécution résultera en un SEGFAULT dû à l'EPERM. Afin de le faire fonctionner juste en modifiant la compilation, il nous faut ajouter l'option « --static » lors de la compilation. Notre programme pourra maintenant fonctionner correctement.

Le code empêchant les syscall depuis le code utilisateur se trouve dans le fichier « /sys/sys/syscall_mi.h ». Dans ce fichier on trouve, ligne 95, une condition impliquant que le PC ne doit pas être sur du code « écrivable ». Il suffit donc de supprimer cette condition et de recompiler notre kernel pour que notre programme fonctionne sans avoir à compiler avec l'option « --static ».