

Enigme système : niveau 6

18 janvier 2017

1 Le background

Les novices en Unix se demandent souvent comment ils peuvent changer leur mot de passe eux-mêmes alors que le fichier dans lequel ils sont stockés (`/etc/shadow`) est illisible à moins d'être `root`. Normalement, quand on exécute une commande, c'est avec ses propres droits. Donc quand un utilisateur lambda lance `passwd`, c'est avec ses droits et il ne devrait pas pouvoir modifier le fichier des mots de passe. C'est sans compter un droit spécial, le bit `setuid`, qui permet d'exécuter une commande avec les droits de son propriétaire. En l'occurrence `passwd` appartient à `root` et son bit `setuid` est positionné, d'où l'accès possible à `shadow`.

Bien entendu, il n'est pas très sécuritaire que `shadow` s'exécute en permanence avec les droits `root`. Imaginons que nous puissions détourner la commande de son but. On pourrait alors lire ou modifier les fichiers de `root`. C'est pourquoi, les commandes dont le bit `setuid` est positionné contiennent des instructions pour "perdre" leur privilèges pour faire des opérations triviales et les regagner quand il y en a vraiment besoin (cf. fonction C `seteuid()`). Par exemple, `passwd` n'est réellement avec les droits `root` qu'au moment d'écrire dans `shadow`.

Le problème, c'est de connaître les façons de programmer correctement pour regagner les privilèges sans causer de faille exploitable. Il y a plein d'exemples historiques contenant une faille, qui ont permis à des hackers de piquer des informations normalement non lisibles. C'est le cas pour cette énigme qui vous emmène à l'orée du monde du vrai hacking.

2 L'énigme

Connectez-vous sur le serveur, à partir d'une machine de l'IUT : `ssh level16@domjudge`

Vous tapez le mot de passe trouvé à l'énigme 5. Vous êtes alors logué avec comme répertoire courant la racine / du système de fichiers.

L'exécutable `/usr/bin/xchgpas` est une commande qui vous permet de changer les mots de passes contenus dans `/etc/secretpass`. Ce fichier et la commande appartiennent tous les deux à l'utilisateur `level17` (on a pas mis `root` pour des raisons évidentes) et bien entendu, seul ce dernier peut consulter le contenu de `secretpass`.

Tout ce que vous savez, c'est la façon d'utiliser la commande. Pour cela, vous devez créer un fichier texte contenant $N + 1$ lignes. La première contient simplement le nombre

N. Sur chaque ligne suivante, il y a du texte au format `login:new_password`.

Par exemple, ce fichier texte peut contenir :

```
2
alice:azertyui
bob:qsd fghjk
```

Pour chaque ligne avec un login correspondant dans `secretpass`, le mot de passe va être changé, avec un message d'information résumant le changement. Pour les logins sans correspondance, il ne se passe rien.

Comme le fichier `secretpass` vous est inaccessible, vous ne pouvez donc connaître les logins qui sont présents dedans. Si vous surmontez cette difficulté, vous trouverez le mot de passe de `level7`.

Indice 1 : `strace` est ton ami pour déterminer (au moins partiellement) quels appels systèmes sont fait par un programme et dans quel ordre.

Indice 2 : « race condition »

Indice 3 : « faire prendre des vessies pour des lanternes »

3 les ressources

Pour vous aider dans la réalisation du programme, vous trouverez sur

<http://cours-info.iut-bm.univ-fcomte.fr>

un article dans la section `hackaton` → édition 2017, portant le même titre que l'exercice. Il contient un lien permettant de télécharger un canevas de code permettant de comptabiliser la solution.