Rootkit - Documentation

Aperçu

Ce rootkit est conçu pour s'intégrer dans le noyau Linux, permettant de cacher des processus et de logger les frappes de clavier.

Il intercepte certains appels système et manipule les processus et les événements du clavier pour en masquer certains ou pour en enregistrer d'autres à des fins d'analyse.

Fonctionnalités

1. Masquage de processus :

Le rootkit intercepte l'appel système `getdents64`, utilisé pour lire les répertoires, afin de cacher les processus dont le nom contient "hidden_process". Cela permet de rendre invisibles certains processus à l'utilisateur via des commandes telles que `ls`.

2. Keylogger (Journalisation des frappes clavier) :

Le rootkit enregistre les frappes de clavier à l'aide du notifier du clavier, qui est une fonction permettant de recevoir les événements du clavier. Chaque touche pressée est loggée dans un tampon. Les frappes peuvent ensuite être extraites ou analysées.

Manuel Utilisateur

Installation :

1. Cloner ou télécharger le code source :

Téléchargez le code source du rootkit ou clonez-le à partir du dépôt.

2. Compiler le module :

Utilisez la commande suivante pour compiler le module kernel :

make

3. Charger le module :

Une fois compilé, vous pouvez charger le module dans le noyau avec `insmod` :

sudo insmod rootkit.ko

Si le module est chargé correctement, vous devriez voir un message de confirmation dans les logs du noyau (`dmesg`).

Fonctionnalités disponibles après installation :

1. Masquage des processus :

Après l'installation du rootkit, tout processus dont le nom contient `hidden_process` sera caché dans la sortie de `ls` et autres commandes similaires qui parcourent les répertoires système.

Exemple : Si un processus appelé "hidden_process" est en cours d'exécution, il sera invisible à l'aide des commandes comme `ps`, `top`, `ls` dans `/proc`, etc.

2. Keylogger :

Le rootkit enregistre toutes les frappes de clavier. Le tampon de keylogger est rempli à chaque fois qu'une touche est pressée, et les valeurs sont stockées dans le tableau `keystroke_buffer`.

Extraction des frappes :

Le rootkit ne prévoit pas d'extraction directe dans le code fourni, mais les frappes peuvent être récupérées en accédant à la mémoire via un outil ou un script externe. C'est une fonctionnalité potentiellement dangereuse et il est conseillé de l'utiliser uniquement dans un environnement de test sécurisé.

Désinstallation

Retirer le rootkit :

Pour retirer le rootkit du noyau et annuler toutes les modifications qu'il a apportées, suivez ces étapes :

1. Décharger le module :

Utilisez la commande `rmmod` pour décharger le module du noyau :

sudo rmmod rootkit

Si cette commande échoue, il est possible que le module soit encore utilisé par des processus. Dans ce cas, utilisez `lsmod` pour vérifier si le module est toujours actif.

2. Vérifier l'état du module :

Après avoir retiré le module, vous pouvez vérifier qu'il a bien été supprimé avec :

lsmod | grep rootkit

Considérations de sécurité

Le rootkit peut être dangereux s'il est utilisé dans un environnement de production ou un environnement où la sécurité est une priorité.

Ce module cache des processus et capture des frappes de clavier, ce qui pourrait être utilisé à des fins malveillantes.

Risque de manipulation de données sensibles :

Les frappes de clavier peuvent contenir des informations sensibles, telles que des mots de passe et des commandes d'administration.

Il est essentiel de tester ce type de rootkit uniquement dans un environnement isolé, comme une machine virtuelle dédiée, afin de limiter tout risque de compromission de données.

Code Source

Le code source du rootkit est divisé en deux parties principales :

1. Modification de `getdents64` :

Le rootkit redéfinit la fonction `getdents64` pour masquer des processus.

Cette fonction est utilisée pour lire les fichiers et répertoires. Le rootkit filtre la sortie pour ignorer les processus portant le nom "hidden_process".

2. Keylogger :

Le rootkit enregistre les frappes du clavier en utilisant la fonction de notification clavier de Linux.

Il capture les touches pressées et les stocke dans un tampon pour une utilisation future.

Sources des fonctions utilisées

- `getdents64` : Cette fonction permet de lire le contenu d'un répertoire. Elle est interceptée pour masquer certains processus.
- `keyboard_notifier_param` : Cette structure permet d'écouter les événements de frappe clavier.
- `sys_call_table` : Utilisée pour modifier directement les appels système.
- `kallsyms_lookup_name` : Permet de trouver des symboles dans le noyau (comme `sys_call_table`).