Structure mémoire du noyau

Matthieu Herrb

CNRS-LAAS

25 septembre 2007

Plan

Plan

Introduction

Le noyau : partie centrale d'un système.

- Connait l'activité et la configuration du système.
- Privilèges les plus élevés sur la machine.
- Point d'accès privilégié pour l'analyse à chaud d´une machine.

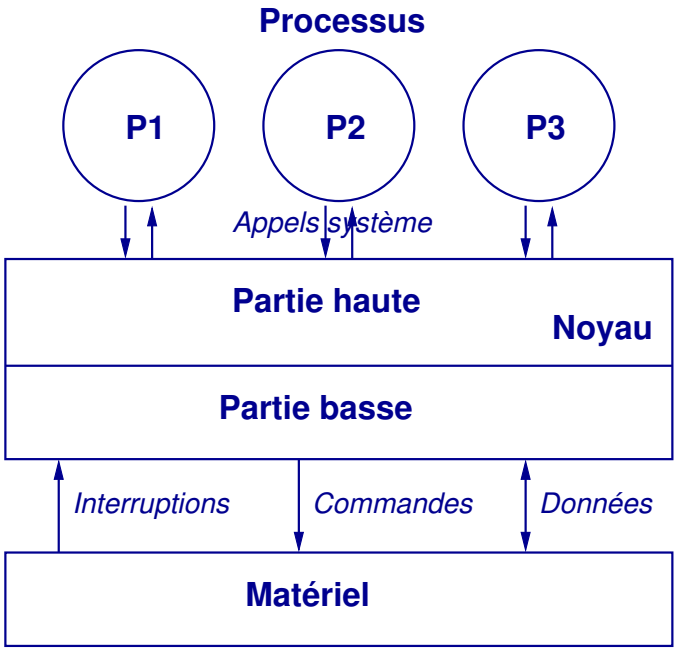
Mais aussi cible ultime des attaques...

Plan

Rôle

- Gestion de la machine : démarrage, arrêt,
- Exécution des applications : gestion des processus,
- Interface avec le matériel : gestionnaires de périphériques,
- Système de fichiers,
- Pile(s) réseau,
- Sécurité : contrôle d'accès aux ressources, gestion des erreurs fatales.

Fonctionnement



Le code du noyau comporte deux parties :

- la partie «haute » exécutée par chaque processus dans son propre contexte (appels système)
- la partie « basse » exécutée de façon asynchrone pour traiter les évènements matériels (interruptions) : arrivée de paquets réseau, clavier, souris, données du disque, etc.

Structures de données

Deux structures importantes :

- la **table des processus**
- la **table des fichiers**

Processus / Threads

Un **processus** = un programme en cours d'exécution.

- Code/données partagés,
- Code/données privés.
- Pile d'exécution.

Pour chaque processus le noyau enregistre :

- son nom, la commande et les paramètres utilisés
- son propriétaire et son groupe
- des infos sur les zones mémoire qu'il utilise
- les descripteurs de fichiers ouverts
- etc.

Threads : mécanisme qui permet d'avoir plusieurs fils d'exécution dans un processus. Partagent le code et les données du processus, mais piles séparées.

Appels système

Mécanisme par lequel un processus demande un service au noyau.
Exemples :

- `open`, `read`, `write`, `close` : transferts de données (fichiers, réseau, etc.)
- `fork`, `exec` : création et exécution de nouveaux processus
- `chdir`, `getdirentries`, `mkdir`, `rmdir` : manipulation du système de fichiers
- `getuid`, `access`, `chmod`, `chown` : manipulation des droits d'accès.

Outils : **`strace`** (Linux), **`truss`** (Solaris), **`ktrace/kdump`** (BSD).

Descripteurs de fichiers

Philosophie Unix : **Tout est fichier**

- données dans le système de fichier
- gestionnaires de périphériques
- ports de communication inter-processus (*pipes*)
- ports de communication réseau (*sockets*)
- etc

Tous les descripteurs de fichiers sont stockés dans une table du noyau.
$\longrightarrow$ Observer cette table donne beaucoup d'information sur l'activité du système.

Modules

Mécanisme permettant d'ajouter dynamiquement des fonctionnalités au noyau :

- gestionnaires de périphériques
- appels système
- etc.

Un module peut également modifier le comportement de sous-systèmes existants :

- corriger un bug
- ajouter des options
- mais aussi : cacher des informations, ouvrir des portes dérobées :
  $\longrightarrow$ rootkits « noyau » .

Plan

Procfs

Un système de fichier spécial :

- crée une hierarchie (`/proc/`) de « fichiers » représentant l'état du noyau
- lecture/écriture pour certains, lecture seule pour d'autres
- mélange entre données formattées pour lecture avec 'cat' ou 'more' et données binaires pour outils externes

Exemples :

- `/proc/modules` liste des modules
- `/proc/cpuinfo` information sur le processeur

lsof

Commande qui permet d'accéder à la table des fichiers ouverts du noyau :

- fichiers sur disque
- connexions réseau

Pour chaque descripteur retourne des informations :

- Object concerné (fichier, socket, etc.)
- processus
- Utilisateur
- Opérations en cours

autres commandes

netstat	liste les connexions réseau et les sockets Unix
ps	commande traditionnelle pour lister la table des processus.
kdump	collecte d'images mémpoire du noyau
crash	analyse de l'image mémoire du noyau

Plan

Les traces d'un pirate

- fichiers ouverts
- connexions réseau
- modules noyau
- ...

Mais : si le pirate a acquis le contrôle du noyau, il peut effacer (presque) toutes ses traces en interceptant les appels système utilisés par les fonctions d'introspection.
Dernier recours : désassemblage de la mémoire du noyau pour trouver ce code de masquage des appels système.

Plan

Conclusion

- Le noyau est la partie la plus privilégiée du système.
- Il existe des vulnérabilités qui permettent d'atteindre le noyau
- Si le pirate n'a pas atteint cette partie, il y laisse des traces visibles
- Si il a atteint cette partie, il peut devenir très difficile à détecter
- L'analyse peut se faire à chaud ou à froid à partir d'une image mémoire sur disque (kcore).