# Sécurité des Systèmes d'Exploitation

Jérémy Briffaut, Christian Toinard

Le but de ce cours est de faire un état de l'art des techniques et des travaux existants dans le domaine de la protection des systèmes d'exploitation. Il vise d'une part à définir les objectifs et les fondements de la protection d'un système et d'autre part à couvrir les systèmes typiques (Unix, Windows) en montrant les approches classiques et les garanties offertes par les différentes méthodes de protections.

1. Introduction

   Le but est d'introduire l'intérêt d'une protection système et les concepts sur lesquelles elle repose.

   1.1 Objectifs
   - 1.1.1 Garantie de confidentialité et d'intégrité
   - 1.1.2 Confinement des vulnérabilités
   - 1.1.3 Contrôle des flux entre les ressources

   1.2 Concepts
   - 1.2.1 Minimisation des privilèges
   - 1.2.2 Contrôle d'accès aux ressources
   - 1.2.3 Flux directs
   - 1.2.4 Flux indirects
   - 1.2.5 Protection discrétionnaire (Discretionary Access Control)
   - 1.2.6 Protection obligatoire (Mandatory Access Control)
   - 1.2.7 Propriétés de sécurité (confidentialité, intégrité)

2. Résultats fondamentaux

   On présente les résultats d'impossibilité en matière de protection ainsi que les difficultés pour avoir des garanties et contrôler tous les niveaux d'un système d'exploitation (interface graphique, application, processus, noyau, matériel, réseau, …).

   2.1 Impossibilité de la protection discrétionnaire
   2.2 Difficultés des approches mandataires
   2.3 Difficultés d'une approche en profondeur
   - 2.3.1 Contrôle des flux dans les interfaces graphiques (flux graphiques, transferts via l'interface graphique, …)
   - 2.3.2 Contrôle des flux dans les applications (caractérisation d'un flux applicatif, domaine d'application, politiques dynamiques, …)
   - 2.3.3 Contrôle des flux dans les programmes (binaires, sources, classes, …)
   - 2.3.4 Contrôle des flux avec le noyau du système d'exploitation (appel système, politiques obligatoires, …)
   - 2.3.5 Contrôle des composants matériels (MMU, carte réseau, carte graphique, …)
   - 2.3.6 Contrôle des flux réseau (flux entre les processus et les interfaces, filtrage dynamique, …)

3. Approches existantes

L'objectif est d'une part de présenter les différents modèles de protection qui définissent des approches conceptuelles pour traiter des besoins spécifiques de protection et d'autre part de classifier les approches à vocation plus générale en allant de cas particuliers pour aller vers des moyens génériques permettant de garantir des objectifs de sécurité assez larges.

3.1 Modèles traditionnels
- 3.1.1 Protection discrétionnaire
  - 3.1.1.1 Unix
  - 3.1.1.2 Windows

3.2 Modèles de protection
- 3.2.1 Protection obligatoire
  - 3.2.1.1 Moniteur de référence système
  - 3.2.1.2 Moniteur de référence applicatif (.NET 3.5 et 4.0 , Java, …)
- 3.2.2 Bell et Lapadula
- 3.2.3 Biba
- 3.2.4 Muraille de chine
- 3.2.5 Définition de domaines et de types

3.3 Classification des approches
- 3.3.1 Approche par automate
- 3.3.2 Séparation par niveau
- 3.3.3 Base d'exécution
- 3.3.4 Gestion des privilèges
  - 3.3.4.1 Abus de privilèges
  - 3.3.4.2 Séparation des privilèges
  - 3.3.4.3 Contrôle de la concurrence
    - 3.3.4.3.1 Conditions de concurrence
    - 3.3.4.3.2 Non-interférence
- 3.3.5 Contrôle des flux
  - 3.3.5.1 Coloration (principe et faiblesse)
  - 3.3.5.2 Analyse de politique (analyse de politiques directes, analyse SELinux, …)
  - 3.3.5.3 Formalisation des besoins de protection (langage de description, logique sur les flux systèmes, PIGA-Security Property Language)

4. Cas pratiques

L'idée ici est de se concentrer sur deux cas pratiques, d'une part SELinux et PIGA pour montrer l'efficacité des approches obligatoires directes et orientées besoins de sécurité et d'autre part Windows afin de montrer le principe du modèle orienté niveau de Windows 7 et Vista ainsi que l'approche obligatoire de .NET 3.5 et 4.0.

4.1 Unix
- 4.1.1 Droits classiques
- 4.1.2 SELinux
  - 4.1.2.1 Principe
  - 4.1.2.2 Définition de la politique d'une application
  - 4.1.2.3 Déploiement sur un système
  - 4.1.2.4 Test et vérification de la protection
- 4.1.3 PIGA
  - 4.1.3.1 Principe
  - 4.1.3.2 Définition d'objectifs de protection
    - 4.1.3.2.1 Canevas
    - 4.1.3.2.2 Propriété
  - 4.1.3.3 Déploiement sur un système
  - 4.1.3.4 Test et raffinement des garanties

4.2 Windows
- 4.2.1 Droits classiques
- 4.2.2 Niveaux d'intégrité
  - 4.2.2.1 Principe
  - 4.2.2.2 Gestion des niveaux d'intégrité
  - 4.2.2.3 Déploiement sur un système
  - 4.2.2.4 Test et vérification
- 4.2.3 .NET
  - 4.2.3.1 Historique
  - 4.2.3.2 .NET 3.5
    - 4.2.3.2.1 Gestion des politiques obligatoires
    - 4.2.3.2.2 Déploiement sur un système
    - 4.2.3.2.3 Test et mise au point
  - 4.2.3.3 .NET 4.0
    - 4.2.3.3.1 Gestion des bacs à sables
    - 4.2.3.3.2 Déploiement sur un système
    - 4.2.3.3.3 Test et mise au point