Linux Security Modules

Sicherheit als Kernelmodul

Michael Pradel

michael@binaervarianz.de

Inhalt

1. Einleitung
2. Sicherheitsmodelle
3. Das LSM Framework
4. Zusammenfassung

Einleitung

- zahlreiche Anwendungen mit hohem Sicherheitsanspruch
- 3 Hauptziele: Datenvertraulichkeit, Datenintegrität, Systemverfügbarkeit
- einige Schwachpunkte von Linux bekannt (mehr dazu später)
- flexible Einbindung verschiedener Sicherheitsmodelle über Module: LSM

Sicherheitsmodelle - ACL

- Darstellung von Zugriffsrechten von Subjekten auf Objekte in Schutzmatrix
- Zugriffskontrolllisten (ACL) unter Linux:
  9 Bit Tupel, z.B. `rwx-x-x`
  - Problem 1: nicht alle Zugriffsrechte ausdrückbar
  - Problem 2: eingeschränkte Rechte nicht möglich, z.B. Schreiben der Passwortdatei
  - Lösungsversuch: SETUID
    - neues Problem: volle Root-Rechte bei fehlerhaften Programmen
    - nicht allgemein genug

Sicherheitsmodelle - Capabilities

- Capability ist Recht, an einem Objekt eine Operation auszuführen
- zuteilbar und wieder entziehbar
- unter Linux:
  - normalerweise Alles-oder-Nichts (Root oder User)
  - 28 Capabilities definiert, z.B. `CAP_CHOWN`, `CAP_SYS_MODULE`
  - Entziehung von root-Rechten bis zum nächsten Neustart mit *lcap* (z.B. *lcap CAP_CHOWN*)
  - keine Rechte an bestimmten Objekten, sondern allgemeiner $\rightarrow$ somit z.B. nicht auf bestimmte Dateien anwendbar

Sicherheitsmodelle - DAC und MAC

- Discretionary Access Control (DAC):
  - benutzerbestimmt
  - normaler Weg unter Linux
- Mandatory Access Control (MAC):
  - systembestimmt, d.h. durch zentrale Stelle
  - Einschränkungen "von oben" können von Benutzer höchstens verstärkt werden
  - z.B. Multi Level Security (Zuordnung von Sicherheitsstufe zu jedem Subjekt und Objekt, Zugriff nur bei ausreichender Stufe)
  - mit SELinux realisierbar

Das LSM Framework - Allgemein

- Bedarf nach umfangreicheren Sicherheitsmodellen
- Implementierung im Kern für direkten Zugriff auf Kernobjekte erforderlich $\rightarrow$ eine (problematische) Variante: Patchen des Kerns
- Realisierbarkeit als Modul angestrebt
- 3 wichtige Forderungen:
  - allgemein, um verschiedene Modelle zu unterstützen
  - Eingriffe in Linux-Kern möglichst klein
  - Implementierung der bekannten Capabilities als Modul

Das LSM Framework - Sicherheitsfelder

- Modul benötigt Möglichkeit, sicherheitsrelevante Informationen mit Kernobjekten zu verknüpfen und zu speichern
- Hinzufügen von Sicherheitsfeldern durch Pointer, die vom Rest des Kerns ignoriert werden, zu ausgewählten Kernobjekten
- z.B. `task_struct`, `inode`, `file`, `net_device`
- Verwendung und Verwaltung der Felder allein Modul überlassen
- Problem bei mehreren Modulen, da kein Kooperationsmodell definiert

Das LSM Framework - Hook Functions

- Modul benötigt Möglichkeit, sich an kritischen Stellen im Kern "einzuhaken"
- Hook Functions können vom Modul implementiert werden und haben direkten Zugriff auf Kernobjekte
- z.B. beim Erstellen und Löschen von Kernobjekten, um Sicherheitsfelder mit Anfangswerten zu belegen
- Beispiel: Erstellen eines Verzeichnisses im virtuellen Dateisystem vfs $\rightarrow$ *Quelltext*
- Limitierung von LSM: (fast) nur Einschränkungen der Rechte möglich

Zusammenfassung

- standardmäßig in Linux integrierte Sicherheitsmodelle für höhere Ansprüche nicht ausreichend
  - z.B. Gefahr durch SETUID, kein MAC möglich
- Schaffung einer generischen Möglichkeit, Sicherheitsmodelle in Form von Modulen einzubinden durch *Linux Security Modules*
- Modul kann u.a. Sicherheitfelder belegen und Hook Functions implementieren
- als LSM implementiert: Capabilities, SELinux, u.a.

Danke für die Aufmerksamkeit! Fragen?

Quellen:

- *Moderne Betriebssysteme* - Andrew Tanenbaum
- *Linux Security Modules: General Security Support for the Linux Kernel* - Chris Wright, Crispin Cowan
- *Integrating Flexible Support for Security Policies into the Linux Operating System* - Peter Loscocco, Stephen Smalley
- *Implementing SELinux as a Linux Security Module* - Stephen Smalley
- Linux Source 2.6.2