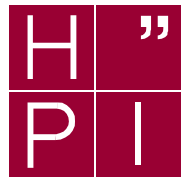
RSBAC

Who is root Anyway ?!

Johannes Nicolai

Seminar Betriebsystemdienste und Administration - SS 2005



Gliederung

- Was sind die zu lösenden Probleme?
- Was ist RSBAC?
- Wie funktioniert RSBAC?
- Welche Module enthält RSBAC?
- Beispiele für die Benutzung von RSBAC
- Überblick über ähnliche Systeme
- Zusammenfassung
- Quellen

Was sind die zu lösenden Probleme?

- **Designprobleme der Linux Berechtigungsarchitektur:**
  - **Allmacht von root, Ohnmächtigkeit normaler Nutzer:**
    - viele Aktionen darf nur root ausführen (ping, apache)
    - Rechtedelegation kaum möglich
    - root darf alles, ist nicht einschränkbar
    - Aktionen von root können nicht zurückverfolgt werden (auditing)
  - **fehlende Granularität:**
    - Rechte setzen am Nutzer und nicht am Programm an
    - rwx Bits sehr unflexibel, Ressource kann nur einer Gruppe gehören
    - Nutzer können Rechte eigener Ressourcen stets selbst verwalten
    - Netzwerkverbindungen, IPC kaum einschränkbar

Was sind die zu lösenden Probleme?

- **Häufig folgende Konsequenzen:**
  - priviligierte Aktion wird benötigt, Programm benötigt root – Rechte
  - Programm benötigt suid Bit oder wird nur von root ausgeführt
  - eingeschleuster Code kann beliebigen Schaden anrichten

  - Nutzerverwaltung in doppelter Hinsicht problematisch:
    - Nutzer geben aus Unkenntnis ihre Dateien allen frei
    - Nutzer können kaum gezielt anderen Rechte vergeben

  - Administration des Systems nur über root- Account
  - kein Schutz vor fehlerhaften Eingaben
  - Aufklärung durch fehlendes Auditing kaum möglich

Was ist RSBAC?

- **RSBAC: Rule Set Based Access Control**
- **Ziele / Anforderungen:**
  - **flexibles, wirksames, feingranulares Zugriffskontrollkonzept, welches Linux Mechanismen ergänzt**
  - **Macht von root kann wirksam eingeschränkt werden**
  - **Einbindung von verschiedenen Entscheidungsmodellen möglich**
  - **Datenmodell und Persistenz unabhängig vom Entscheidungsmodell**
  - **Trennung zwischen Komponenten, die Entscheidung durchsetzen und Komponenten, die Entscheidung fällen**
- **RSBAC kann alle skizzierten Probleme lösen**

Was ist RSBAC?

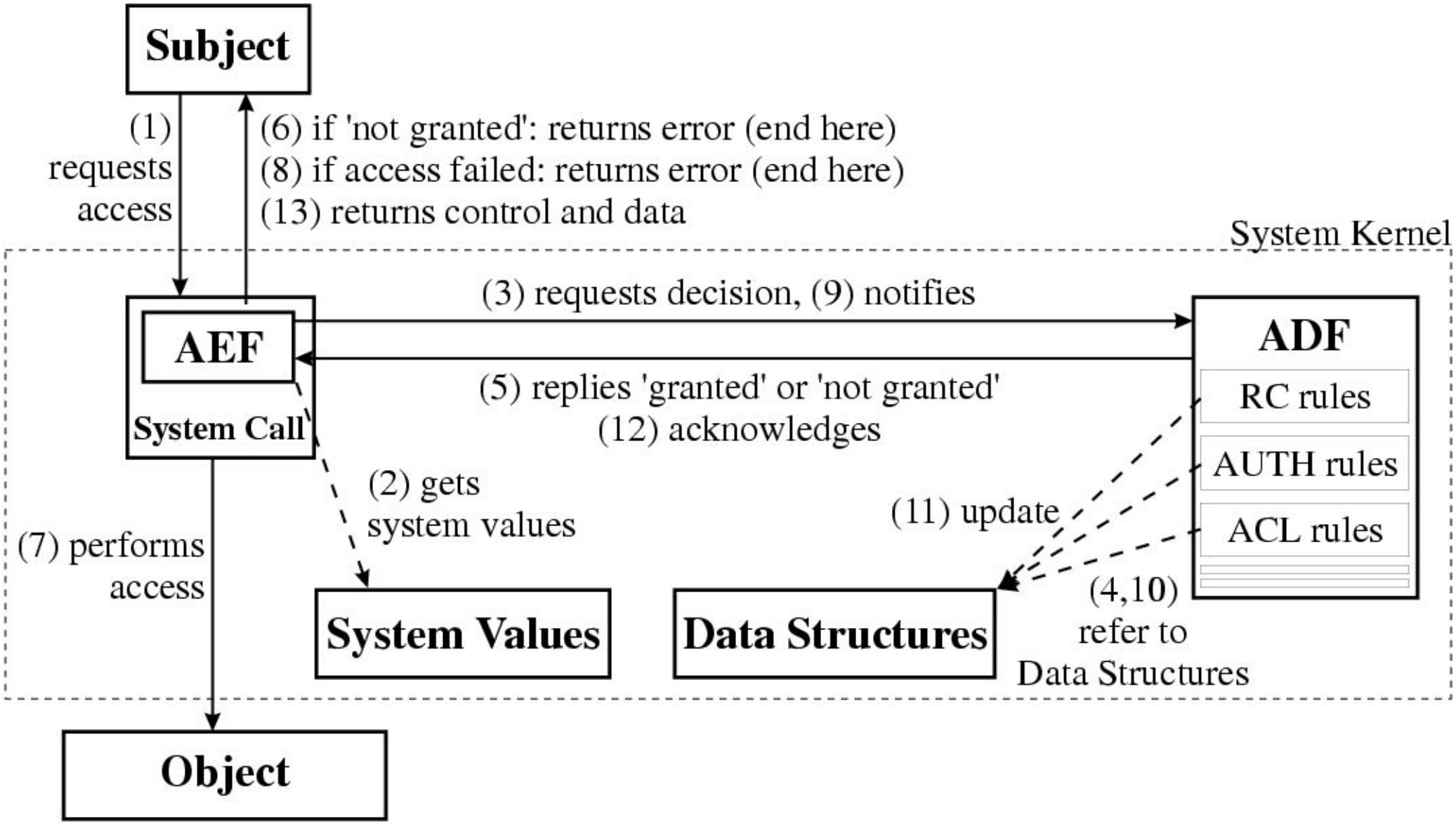
- kernel patch + user land Tools (Kommandozeile und menübasiert: rsbac_menu)
- Entwickler: Amon Ott, Lizenz: GPL
- Diplomarbeit im November 1996
- stabil seit März 2000
- aktuelle Version: 1.2.4

- Distributionen mit RSBAC Support
  - Sniffix (Knoppix Life CD)
  - Adamantix (zur Zeit große Umstellungen)
  - Gentoo Hardened
  - ALTLinux Castle, Kaladix (beide veraltet)

Wie funktioniert RSBAC?

- Subjekt greift mit Request auf Objekt zu
- Subjekte:
  - Nutzer, die ein Programm ausführen
  - Programme
- Objekte:
  - files, directories, fifos, symlinks, devices, ipcs
  - users, processes
  - netdevs, nettemps, netobjs
- Requests:
  - 46 verschiedene Arten, sehr feingranular
  - wird in Beispielen noch ersichtlich

Wie funktioniert RSBAC?



Wie funktioniert RSBAC?

- **Implementation:**
  - **Modulregistration zur Laufzeit**
  - **Modifikation von Systemrufen**
  - **Einführung neuer Systemrufe**
  - **zentrale Datenhaltung und modellspezifische Datenhaltung**
    - **automatisches SMP locking**
    - **Ablage in generischen, mehrstufigen Listenformat**
    - **Persistenzhaltung: daemon rsbacd**
    - **Lebensdauer von Attributen erlaubt weitere Granularität**
  - **Security Officer**
    - **default UID 400**
    - **darf RSBAC Attribute ändern, keine weiteren Privilegien**

Welche Module enthält RSBAC?

- **Module, die in Beispiel nicht behandelt werden:**
  - **MAC: Mandatory Access Control nach Bell- La Padula**
  - **FC / SIM: Functional Control: Zugriffslevel für jede Ressource**
  - **PM: Datenschutzrichtlinien nach Simone Fischer Hübner**
  - **MS: Ausführen eines Virenscanners vor Lesen und Ausführen**
  - **JAIL: sperrt Prozesse in einen Käfig ein**
  - **RES: gezielte Beschränkung von Ressourcen von Prozessen**
  - **PAX: verhindert große Anzahl von Stack- / Heap Smashing Attacks durch Non Executable Stacks / Verhinderung von Race Conditions und Address Layout Randomization (siehe Root Kits Vortrag)**
  - **USER: Nutzerverwaltung komplett im Kernel**
  - **LOG: Auditing nach Rolle, Benutzer, Programm, Aktion**

Welche Module enthält RSBAC?

- **Module, die in Beispiel behandelt werden:**
  - **FF: File Flags, erlauben globale Dateiattribute (read- only, no – execute, secure- delete, append-only)**
  - **CAP: Linux Capabilities: priviligierte Aktionen root entziehen, anderen Benutzern erlauben**
  - **AUTH: Unter welchen Benutzerkennungen darf Prozess laufen?**
  - **ACL: Access Control Lists: analog zu Netware ACLs, Gruppenverwaltung für Nutzer möglich**
  - **RC: Role Compatibility**
    - **Subjekte werden zu Rollen zusammengefasst**
    - **Objekte werden zu Typen zusammengefasst**
    - **Zugriffsmatrix Rolle / Typ, Inhalt, Requesttypen**
    - **Aufspaltung von Administrationsaufgaben möglich**

Beispiele für die Benutzung von RSBAC

- kein Ausführen unter /home/
- ping ohne suid root
- kein su zu bob aber zu alice
- Gruppenbeispiel mit ACLs
- root darf nicht in /tmp/ löschen, alle anderen schon
- (Apache darf von jedem gestartet werden, nur er kann Verbindungen auf Port 80 annehmen)

Überblick über ähnliche Systeme

- SELinux: größter Konkurrent von der NSA, wesentlich komplizierter, noch Prototyp
- LIDS: statischer als RSBAC, Möglichkeiten nicht so zahlreich
- BSD Jails: funktionsärmer als JAIL Modul
- Systrace: benutzerdefinierte Policy Engine im User Space, wesentlich einfacher, nur für einzelne Programme geeignet
- Windows NT: teilweise bessere Granularität als Linux, viele Probleme bleiben aber bestehen
- detaillierte Übersicht unter http://myhpi.de/~nicolai/

Zusammenfassung

- RSBAC: flexibles, wirksames, feingranulares Zugriffskontrollkonzept, welches Linux Mechanismen ergänzt
- Macht von root kann wirksam eingeschränkt werden, Delegation von Rechten möglich
- Rechte eines Programms können exakt an seine Aufgabe angepasst werden
- klare Trennung zwischen Durchsetzungskomponente, Entscheidungskomponenten und Datenhaltungskomponenten
- sehr hohe Komplexität durch große Anzahl von Modulen
- Lizenz: GPL
- Distributionen: adamantix, sniffix, gentoo hardened

Quellen

- http://www.rsbac.org
- Amon Ott, “Root ist nicht alles – Mehr Sicherheit für Linux-Server mit RSBAC”, iX 8/02, S. 99
- Amon Ott, “Die Modelle des Linux- Sicherheitssystems RSBAC”, Linux- Magazin 04/03, S. 61 – 67
- Amon Ott, “Die Architektur des Linux- Sicherheitssystems RSBAC”, Linux- Magazin 01/03, S. 48 – 51
- http://myhpi.de/~nicolai/