SELinux

Workshop auf dem LinuxInfoTag 2003
in Dresden

Carsten Grohmann <mail@carstengrohmann.de>
18. Oktober 2003

Agenda

I. Kleine SELinux-Kunde

- Erweiterung für Linux
- feinere Zugriffssteuerung
- Programme laufen in Domänen

II. Vorführung eines SELinux-Systems

III. Praxis: Grafische Tools zur Policy-Verwaltung

Begriffe

- Security Context (Benutzer:Rolle:Typ/Domäne)
- SID/PSID
- TE Type Enforcement
- RBAC Role Based Access Control
- MAC Mandatory Access Control
- DAC Discretaionary Access Control
- Typen / Domänen und Rollen
- Objekt/Subjekt

Elemente

- RBAC (role based access control)
- Security Context (Benutzer:Rolle:Typ/Domäne)
- Jedes Objekt und jedes Subjekt hat einen Security Context
- Berechtigungen abhängig von
  - Authorisierten Rollen
  - Domänen in der Rolle
  - Wechsel zwischen zwei Rollen oder zwei Typen
- Erzwingen der Berechtigungen

Funktionsweise

- Regelbasiertes System mit TE, RBAC und MAC
- Alle Prozeße laufen in eigenen Domänen
- Policy autorisiert:
- Wechsel zwischen Typen (Domänen)
- Zugriff auf alle Typen (auch die eigenen)
- Wechsel nur bei Start eines Programms möglich
- Berechtigungen sind parallel zu Linux

Betriebsmodi

- Permissive Mode
  - Gewährung aller Zugriffe
  - Protokollierung aller Verstöße
- Enforcing Mode
  - Policy erlaubt und verweigert Zugriffe
  - Protokollierung aller Verstöße

Installation

- Aus den Quellen der NSA(Red Hat 9.0) basiert
- Alternativ Pakete für
  - Debian
  - SuSE
  - Gentoo

Neue Befehle

- chcon, setfiles, load_policy, checkpolicy, runcon
- newrules.pl error2rule
- newrole Rollenwechsel
- Wrapper sichereren Zugriff auf Systemresourcen
  - z.B. schfn, spasswd, schsh und andere

selinuxfs

- eigenes Pseudofilesystem für SELinux
- unter /selinux wie procfs unter /proc
  - enforce SELinux-Modus
  - load Laden der Policy
  - context Überprüfen eines Security Context
  - relabel Dateisystemmarken erzeugen
  - user Benutzerdefinierte Security Context
- nur enforce wird direkt aufgerufen

Vorteile

- Feinere Zugriffsrechte
- Besserer Schutz vor „bösen“ Code
- Allmacht root gebrochen
  - dennoch Henne-Ei-Problem, da ein Admin Regeln ändern kann
- Kein chroot() mehr notwendig

Nachteile

- Höherer Verwaltungsaufwand durch die Policy
- Regeln pro Programmpaket
- Tiefere Systemkenntnisse notwendig
- Erhöhter Installationsaufwand bei Nicht-RH-Distributionen (Ausnahme Debian)
    - Teilweise Probleme beim Patchen / Anpassen der Systemprogramme

Resümee

- Mehr Sicherheit mit vertretbaren Aufwand
- Leicht zu administrieren
- Hürden bei der Installation

Quellen

- http://www.nsa.gov/selinux/index.html
- http://www.securityenhancedlinux.de
- http://sourceforge.net/projects/selinux/