SELinux

Workshop auf dem Chemnitzer LinuxTag 2004

Carsten Grohmann <mail@carstengrohmann.de>
06. März 2004

Agenda

1. Kleine SELinux-Kunde
2. Vorführung eines SELinux-Systems
3. Praxis: Erste eigene Schritte

Workshop Teil 1

- Was ist SELinux?
- Begriffe
- Interna / Funktionsweise
- neue und geänderte Befehle

Was ist SELinux?

- Regelbasiertes System mit TE, RBAC und MAC
- unveränderliche Benutzeridentität über alle Operationen wie su, Rollen- und Domänenwechsel
- feinere Gliederung der Berechtigungen
- bricht mit der Allmacht von root
- Programme laufen in Domänen

Begriffe

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

DAC – Discretionary Access Control

- objektbezogene Zugriffskontrolle
- Einsatz hauptsächlich für Dateien
- Eigentümer setzen die Berechtigungen nach eigenen Ermessen
- Steuerung über funktionale Kriterien -> meist Freigabe von mehr Daten als benötigt
- Beispiele: ACLs, Berechtigungen auf Basis der Benutzer-ID

MAC – Mandatory Access Control

- regelbasierte Zugriffskontrolle
- Steuerung über logischen und funktionale Kriterien -> feiner abgestufte Berechtigungen für alle Ressourcen
- Einsatz bei
  - systemweite Vorgabe der Berechtigungen
  - System erzwingt Entscheidung – auch gehen den Objekteigentümer
- keine Allmacht „root“
- MLS (Multi Level Security) ist ein Spezialfall
- Beispiele: Systrace, LIDS, SELinux

RBAC – Role Based Access Control

- rollenbasierte Zugriffskontrolle :-)
- Berechtigungen werden über die Zuordnung der Benutzer zu Rollen vergeben
- Rolle als Abstraktionsschicht enthält die Berechtigung
- Berechtigungen werden funktionsgebunden vergeben
- Beispiel: RSBAC, SELinux

TE – Type Enforcement

- (Domain and) Type Enforcement
- spezielle Form von MAC
- Zugriffsregelung über Matrix (Tabelle) oder flexibleren Regeln
- Beispiel: SELinux

Objekte und Subjekte

Subjekte: Handelnden Elemente des Systems (Prozesse)

Objekte: Ressourcen (z. B. Dateien, Verzeichnisse, Sockets, IPCs, ...)

Domänen, Typen und Attribute

- Typ: Menge von Objekten und Subjekten mit gleichen Rechten
- Domäne: prozeßgebundener Typ
- Unterscheidung zwischen Domäne und Typ nur sprachlicher Natur, keine internen Unterschiede
- Attribute:
  - Sammlung von Typen
  - Verwendung analog eines Typ
- Typenwechsel erfolgen automatisch gemäß des Regelsatzes

SELinux MAC

- Nachhaltige Trennung von Programmen in Sicherheitsbereiche (Domänen)
- Möglichkeit verdächtige Programme in eigenen Domänen laufen zu lassen
- Domänenwechsel um Programmen nur die notwendige Berechtigungen zu erteilen
- Frage: Welches als root laufende Programm braucht Zugriff auf die persönlichen Daten der Nutzer?

SELinux RBAC

- Rollen sind eine Abstraktionsschicht zur Benutzerverwaltung
- jeden Benutzer sind eine oder mehrere Rollen zugewiesen
- jeder Rolle sind Domänen zugeordnet
- über Rollen wird die Zulässigkeit von Domänen geregelt
- Rollenwechsel nur nach erneuter Authentifizierung via newrole
- Rollenwechsel seltener als Domänenwechsel
- keine automatische Rollenwechsel

SELinux Identität

- der Benutzer von SELinux bleibt immer (auch bei su) unverändert im Gegensatz zum traditionellen Unix
- über die Identität werden die verfügbaren Rollen und damit die nutzbaren Domänen geregelt
- aktuellen Rolle bestimmt Rechte des Nutzers
- im Regelsatz eingetragene Benutzer haben den gleichen Namen wie im Basissystem z. B. carsten -> carsten:user_r:user_t
- unbekannte (nicht eingetragene) Benutzer heißen „user_u“ z.B. bob -> user_u:user_r:user_t

Kennzeichnung von Nutzer, Rollen und Typen

- folgende Suffixe werden verwendet:
  - „_r“ für Rollen
  - „_t“ für Typen und Domänen
  - „_u“ für Standardnutzer wie system_u und user_u

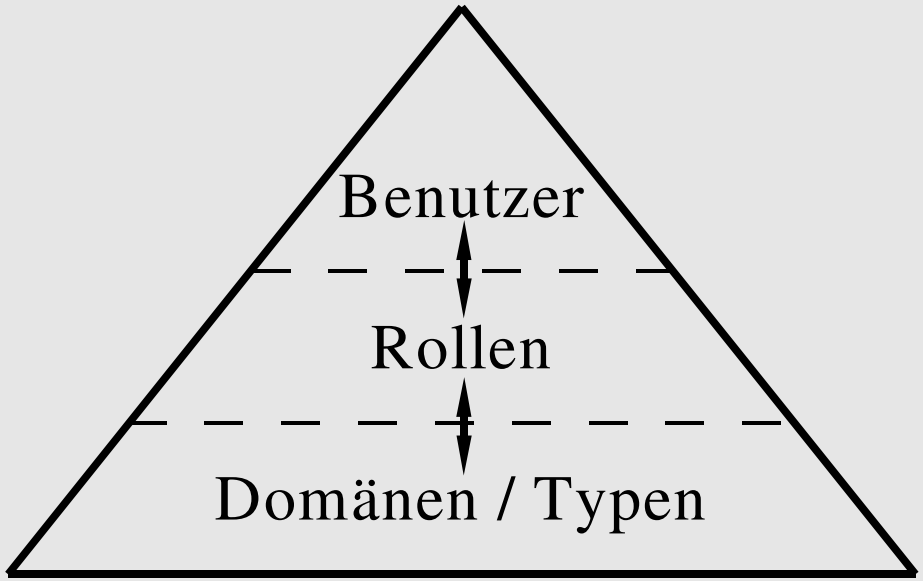
Standardbenutzer / -rollen

- Standardbenutzer:
  - system_u für das System
  - user_u für Benutzer ohne SELinux-Benutzeridentität
- Standardrollen
  - system_r System
  - sysadm_r ehemals root
  - staff_r weniger als root mehr, als unprivilegiert
  - user_r unprivilegierte Nutzer

Security Context

- Darstellung von Benutzer, Rolle und Domäne/Typ als Zeichenkette
- Aufbau Benutzer:Rolle:Typ z. B. carsten:user_r:user_t
- File Context Sicherheitskontext für Dateien
- SID: dynamische Abbildung eines Sicherheitskontext zur Laufzeit als Ganzzahl

Schaubild Security Context



Funktionsweise

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

Funktionsweise (Forts.)

- Jedes Objekt und jedes Subjekt hat einen Sicherheitskontext
- Berechtigungen abhängig von
  - Autorisierten Rollen
  - Domänen in der Rolle
  - Wechsel zwischen zwei Rollen oder zwei Typen
- Erzwingen der Berechtigungen

Sicherheitsmodi

- Permissive Mode
  - Gewährung aller Zugriffe unabhängig vom Regelsatz
  - Protokollierung aller Verstöße
- Enforcing Mode
  - Regelsatz erlaubt und verweigert Zugriffe
  - Protokollierung der ersten Verstöße

Installation

- Voraussetzungen
  - einwandfrei laufendes Linuxsystem vor dem Einspielen der Änderungen
  - Unterstützung der erweiterten Attribute (xattrs)
- Unterstützte Dateisysteme
  - ext2 und ext3
  - XFS
- Quellen der NSA (Fedora Core 1 basiert) oder alternativ Pakete für Debian, SuSE und Gentoo

Bootoptionen

- enforcing=1 startet im enforcing mode
- selinux=0 deaktiviert SELinux

Erweiterte Befehle

- ls (coreutils) und ps (procps) um den Sicherheitskontext von Dateien bzw. Prozessen anzuzeigen
- cp, mv (beide coreutils) und logrotate um den Sicherheitskontext zu erhalten
- cron um die cronjobs im richtigen Sicherheitskontext auszuführen
- login, pam_selinux und wdm um den Sicherheitskontext beim Einloggen benutzerspezifisch zu ändern
- star mit Unterstützung zur Sicherung des Dateikontexts

Neue Befehle

- chcon /setfilecon Sicherheitskontext bei Dateien/Verzeichnissen ändern
- setfiles Sicherheitskontext aller Dateien/Verzeichnisse eines Dateisystems ändern
- load_policy Laden und aktivieren des binären Regelsatzes
- checkpolicy SELinux policy compiler
- audit2allow Umwandlung von Fehlermeldungen in Regeln
- newrole Programm zum Rollenwechsel

Neue Befehle (Forts.)

- run_init Start eines Init-Skripts in der Domäne für die Init-Skripte (initrc_t)
- runcon Start eines Programmes in einen wählbaren Sicherheitskontext
- getenforce Anzeige des aktuellen Sicherheitsmodus
- setenforce [0|1] Wahl des Sicherheitsmodus

selinuxfs

- eigenes Pseudodateisystem für SELinux
- unter /selinux wie procfs unter /proc
- möglichst kein direkter Zugriff
- wird automatisch eingehangen (ohne Eintrag in /etc/fstab)
  - enforce les- und schreibbar zur Anzeige und Änderung des Sicherheitsmodus
  - policyver enthält die Version des binären Regelsatzes

Workshop Teil 2 – Vorführung

- erweiterte Befehle
  - ls / --context und ps --context
  - id
- neue Befehle
  - getenforce und setenforce
- gleichbleibende Identität nach su
  - id -> su gentoo -> id
- Zugriff auf geschützte Systemdateien wie /etc/shadow, /proc/kcore
  - cat /etc/shadow; cat /proc/kcore

Workshop Teil 3 – Policy

- Nutzt Verzeichnisstruktur mit einer Datei pro Dienst
- Wichtige Unterverzeichnisse:
  - domains
    - domains/program/unused enthält nicht aktivierte Regeln
    - Aktivieren durch Verschieben nach domains/program
  - macros
  - types
- Binärversion /etc/security/selinux/policy.VersionsNummer

Vorteile

- Feinere Zugriffsrechte
- Allmacht root gebrochen
  - dennoch Henne-Ei-Problem, da ein Administrator Regeln ändern kann
- Kein chroot() mehr notwendig
- Besserer Schutz vor „bösen“ Code
- Höherer Schutz in unsicheren Umgebungen (auch ohne aktuelle Patches)

Nachteile

- Höherer Verwaltungsaufwand durch den Regelsatz
- Regeln pro Programmpaket
- Tiefere Systemkenntnisse notwendig
- Erhöhter Installationsaufwand manchen Distributionen (Mandrake, Slackware, SuSE bei neueren Paketen) durch Probleme beim Patchen / Anpassen der Systemprogramme

Resümee

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

Danksagung

Chris PeBenito für die SELinux LiveCD
Oliver Tennert für eine Auskunft rund um MAC, DAC
und alle die diesen Vortrag ermöglicht haben

Quellen

Offizielle NSA SELinux Seite:
http://www.nsa.gov/selinux/index.html

Deutschsprachige SELinux Seite:
http://www.securityenhancedlinux.de

SourceForge SELinux Projektseite:
http://sourceforge.net/projects/selinux/

SELinux Mailing List:
http://www.nsa.gov/selinux/list.html und
http://www.vegaa.de/mailman/listinfo/selinux-de

SELinux Mailing List Archive:
http://marc.theaimsgroup.com/?l=selinux

Quellen (Forts.)

IRC:	#selinux auf irc.debian.org
Fedora	ftp://people.redhat.com/dwalsh/SELinux
Fedora-Kernel	http://people.redhat.com/arjanv/2.5/
Debian Unstable	http://www.coker.com.au/newselinux/

Debian Woody
http://www.microcomaustralia.com.au/debian/

SELinux Online Dokumentation