



# Einführung in SELinux

*Ralph Angenendt <[ralph@centos.org](mailto:ralph@centos.org)>*

# SELinux und CentOS 5



- Überblick über das alte Securitymodell
- Was ist in CentOS 5 enthalten?
- Policies, Booleans und Module
- Tools zur Interaktion mit SELinux
- Erstellen eines neuen Policymoduls mit den vorhandenen Tools

# Was gab es schon immer?



- `rw-r-xr-x` ist das klassische Modell zur Rechtevergabe
- Es ist ein einfaches Modell, das man Anfängern leicht beibringen kann – `chmod 777` muss nicht sein
- KISS
- Aber ...
- Es ist zu einfach für komplexe Umgebungen



# rwxrwxrwx

- Problematisch in komplexen Setups
  - Kernel 2.6 erlaubt 65535 Gruppen pro User
  - Aber ...
  - Wenn NFS ins Spiel kommt sind es nur noch 16
- Ein kleines Quiz
  - `/var/www/html` gehört dem Nutzer Apache
  - Gruppe `content` darf lesen und schreiben
  - Gruppe `backup` darf nur lesen
  - Lösung?

# Captain ACL zur Hilfe!



- Moderne Dateisysteme kennen Extended Attributes
- In EAs können Metadaten gespeichert werden
- Warum also nicht access control lists?
- Großartig. Jetzt können wir einem Verzeichnis mehr als einen User oder eine Gruppe zuweisen
- Dies hilft beim Modellieren komplexer Setups
- Das Problem der letzten Folie ist lösbar



# Auftritt S E Linux

- Umdenken: Wer darf was wo tun?
- ALT: Nutzer kontrolliert, wer Zugriff auf selbsterstellte Daten hat (eingeschränkt)
- NEU: Mandatory Access System
  - Alles wird mit einem Kontext gekennzeichnet
  - User benötigt Zugriff auf diesen Kontext
  - Ansonsten keine Möglichkeit Dateien zu ändern
  - Kompromittierter Prozess kann nicht auf Daten zugreifen, auf die “andere” zugreifen dürfen  
(rwxrwxrwx)



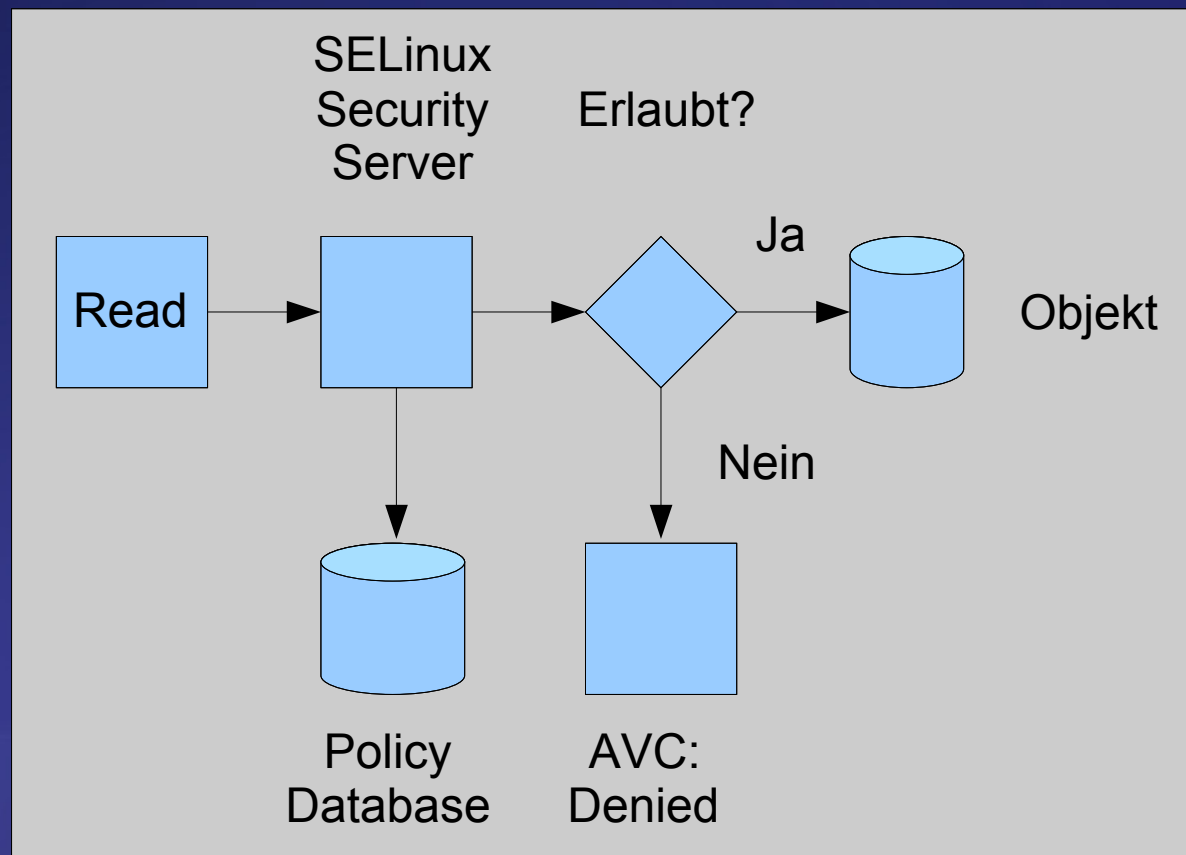
# Whats more?

- SELinux beinhaltet ebenfalls ein RBAC system
  - Zugriffsrechte auf Objekte werden an Rollen gegeben
  - Rollen können entlang dem Betriebskonzept modelliert werden ( Management, Sekretariat, ... )
- Und Multi Level Security
  - Modelliert nach Anforderungen des DOD
  - Unclassified → Confidential → Secret → TOPS
  - Objekte werden klassifiziert, Subjekte bekommen Freigabekriterien



# Und wie funktioniert es?

- Überblick:





# Und wie funktioniert es? (II)



- Drei verschiedene Betriebsmodi
  - Enforcing
  - Permissive
  - Disabled
- Zwei verschiedene Policies
  - strict
  - targeted
  - targeted ist voreingestellt



# Welche Tools haben wir?

- setenforce und getenforce
- chcon
- restorecon
- semodule
- semanage
- ls -Z um Kontexte zu betrachten (ps, id):  
system\_u:object\_r:httpd\_sys\_content\_t
- system-config-selinux



# Und los!

- Beispiel:
  - httpd darf nur in `/var/www/html` lesen
  - DocumentRoot soll in `/data/` sein
  - “`chcon -R --reference=/var/www/html /data`” ändert den Sicherheitskontext von `/data` und den darin liegenden Dateien
  - httpd kann jetzt Dateien aus diesem Verzeichnis ausliefern



# Booleans

- Piffige Möglichkeit um mit der Policy zu interagieren
- Policy muss nicht neu gebaut werden
- `getsebool -a` zeigt alle verfügbaren Booleans
- Beispiel:
  - Nutzer haben Webseiten in `~/public_html/`
  - Management will das nicht mehr
  - `setsebool -p httpd_enable_homedirs off`
  - Voilà. Management ist glücklich



# Weitere Booleans

- `allow_execstack`
- `allow_ftpd_use_cifs`
- `httpd_ssi_exec`
- `samba_share_nfs` (NFS hat keine EAs)
- `httpd_can_network_connect_db`



# SELinux Module

- Neue Regeln in die Policy laden
- Policy muss nicht neu gebaut werden
- audit2allow um neue Regeln zu erstellen
- Liest avc:denied messages
- semodule verwaltet Module (load, unload, update)
- Beispiel: vsftpd soll Verzeichnisse mit httpd\_sys\_content\_t lesen dürfen



# audit2allow

- setenforce=0, starte vsftpd, sammle avc:denied

```
grep vsftpd /var/log/audit/audit.log | audit2allow -m local
module local 1.0;

require {
    type ftpd_t;
    type httpd_sys_content_t;
    class dir { read search getattr };
    class file { read getattr };
}

#===== ftpd_t =====
allow ftpd_t httpd_sys_content_t:dir { read search getattr };
allow ftpd_t httpd_sys_content_t:file { read getattr };
```



# Und jetzt!

- Demotime!
- Fragen!
- Antworten!
- Danke!