Wenn rwx mit ugo zu ungenau ist - SELinux und AppArmor

Jörg Brühe, FromDual GmbH

Inhalt:

Unix-Systeme (also auch Linux) verwalten Zugriffsrechte, die für jedes Objekt (Datei, Verzeichnis, Gerät, ...) dem Eigentümer ("user"), der Gruppe ("group") und allen anderen ("other") das Lesen ("read"), Schreiben ("write") und Ausführen ("execute") erlauben oder verbieten.
Mit dem Kommando "chmod" kann man diese Rechte "r", "w" und "x" für "u", "g" oder "o" gewähren oder entziehen.

Dieses System unterscheidet nur nach Benutzern, aber nicht nach Anwendungen:
Was das Programm X nach Aufruf durch den Benutzer "meier" tun darf, das dürfen auch die Programme Y und Z.

In vielen Fällen ist das sachlich nicht begründet:
Mein Mail-Client muss mein Adressbuch lesen und ändern können, aber warum soll z.B. mein Browser das tun dürfen?
Die Kontrolle der Zugriffe nicht nur nach Benutzer, sondern auch nach Anwendung schränkt den möglichen Schaden eines erfolgreichen Angriffs ein.

Hier setzen die Kernel-Erweiterungen "SELinux" ("Security-Enhanced Linux")und "AppArmor" ("Application Armor") an: Durch sie erhält der Kernel eine "Mandatory Access Control", die alle Zugriffe der Anwendungen prüft und solche verbietet, die in der eingerichteten Sicherheitspolitik nicht vorgesehen sind.
Es handelt sich um Positiv-Listen: Erlaubt ist nur das, was dort eingetragen ist, alles andere ist verboten.
Funktional sind die beiden Erweiterungen ähnlich; Ubuntu bevorzugt AppArmor, RedHat nutzt SELinux.

Dieser Gewinn an Sicherheit bewirkt einen Mehraufwand bei der Administration:
Für Anwendungen, die im System bisher nicht vorgesehen sind (also alle, die nicht von der Distribution geliefert wurden), müssen die erlaubten Zugriffe eingerichtet werden.Das ist auch nötig, wenn die Daten einer Anwendung an einem Nicht-Standard-Pfad abgelegt werden sollen, z.B. eine Datenbank auf einem separaten Filesystem (NAS, RAID, ...) gespeichert werden soll.

Manchen Administratoren ist das zuviel Aufwand: Sie schalten SELinux bzw. AppArmor auf dem System aus. Das erhöht natürlich die Flexibilität und vermeidet Probleme, die durch unvollständige Regeln entstehen (zu erlaubende Zugriffe, die im Test nicht erfasst wurden), aber es reduziert die Sicherheit.

Im Vortrag werden diese beiden Systeme vorgestellt und die grundlegenden Admininistrations-Aufgaben beschrieben:
- den Status der Komponente bestimmen und ändern (an- und ausschalten),
- die eingestellte Politik anzeigen lassen und kontrollieren,
- für erfasste Anwendungen die erlaubten Zugriffe ändern, z.B. wegen der Benutzung von Nicht-Standard-Verzeichnissen.

Sinnvolle Vorkenntnisse:

Privilegien auf Unix-Systemen: ”read“, ”write“, ”execute“
Konzepte ”user“ und ”group“, ”primary group“, ”supplementary group“
Kommandos ”chmod“, ”chown“, ”usermod“