



10 Kritische Betrachtung von AppArmor

10.1 Geschwindigkeit

Die Geschwindigkeit von Mandatory-Access-Control-Systemen bei realen Anwendungen zu messen ist nicht sehr einfach. Natürlich lassen sich die Operationen wie Dateiöffnen und -schließen in ihrer Geschwindigkeit messen, jedoch lassen sich hier von nur sehr geringe Aussagen für spätere echte Anwendungen ablesen. Die Ergebnisse eines entsprechenden Benchmarks finden Sie in Abschnitt 3.1.

Hier wähle ich daher einen anderen Ansatz, der aber auch sicherlich nicht für alle Anwendungen geeignet ist. Die bereits vorgestellte Web-Applikation *phpSysInfo* soll hier genutzt werden. Für eine Analyse der Geschwindigkeit wird diese Applikation mehrfach (1000-mal) ohne und mit AppArmor-Überwachung über ein 100-Mbit/s-Netzwerk aufgerufen. Das getestete System war ein *SUSE 10.1*-System mit sämtlichen Updates auf einem Intel Pentium 4-Rechner mit 3 GHz und 512 Mbyte Arbeitsspeicher. Die Zugriffe erfolgen mit dem Apache-Benchmark-Werkzeug `ab`.

Lauf	Dauer
Ohne AppArmor	238.4s, 239.2, 239.5
Mit AppArmor	242.4s, 244.3, 241.4s

Es lässt sich leicht feststellen, dass die zusätzlichen Tests, die AppArmor durchführt, bei dieser Applikation nicht wesentlich ins Gewicht fallen. Der zusätzliche Overhead liegt bei 1,6%. Dies ist sicherlich noch applikationsabhängig und kann bei anderen Applikationen sowohl nach oben als auch nach unten abweichen. Hier zeigt sich, wie bei vielen anderen Applikationen auch, dass andere Kriterien die Ausführungsgeschwindigkeit beschränken (Netzwerkbandbreite etc.). Der zusätzliche Aufwand für AppArmor ist meist zu vernachlässigen.

10.2 Sicherheit

Wenn Sie bereits bis hierhin gelesen haben, kennen Sie AppArmor bereits recht gut. Falls Sie direkt hierhin geblättert haben, werden Sie vielleicht einige Begriffe nachschlagen müssen. Dennoch sollten Sie in der Lage sein, die folgenden Ausführun-

gen zu verstehen. AppArmor erhöht sicherlich die Sicherheit des Systems, auf dem es installiert wurde. Als Mandatory-Access-Control-System (MAC) muss es sich aber auch den Vergleich mit anderen ähnlichen Lösungen gefallen lassen. Alternative Lösungen sind:

- SELinux
- LIDS (<http://www.lids.org>)
- grsecurity (<http://www.grsecurity.org>)
- RSBAC (<http://www.rsbac.org>)
- etc.

Ich werde hier lediglich AppArmor betrachten. Eine Betrachtung von SELinux erfolgt in dem entsprechenden Kapitel.

Bei der Betrachtung von AppArmor fällt als Erstes die leichte Handhabung, die verständliche Syntax der Profil-Dateien und die gute Dokumentation auf. Dies sind alles Pluspunkte für AppArmor.

Aus Sicht der Sicherheit ist es jedoch bei AppArmor ungünstig, dass auf dem System vertrauenswürdige und nicht vertrauenswürdige Programme unterschieden werden müssen. Die vertrauenswürdigen Programme werden von AppArmor nicht überwacht und dürfen auf dem System jede durch die UNIX-Rechte erlaubte Tätigkeit durchführen. Lediglich die nicht vertrauenswürdigen Applikationen werden von AppArmor zusätzlich überwacht.

Ein zweiter Kritikpunkt ist die Tatsache, dass AppArmor als *Zugriffsattribut* den Dateinamen verwendet. Ein Umbenennen oder Kopieren der Datei entzieht eine Datei oder Applikation der Überwachung durch AppArmor. Änderungen der Installationsorte von Applikationen durch den Programmierer führen dazu, dass die Applikation plötzlich nicht mehr überwacht wird. Außerdem können nicht alle Objekte unter Linux über einen Dateinamen angesprochen werden. Netzwerkverbindungen, Prozesse und IPC-Aufrufe können nicht über Dateinamen angesprochen und daher nicht von AppArmor überwacht werden.

AppArmor kann nicht zwischen unterschiedlichen Benutzern unterscheiden. Applikationen, die auf Benutzerdaten zugreifen müssen, benötigen immer den Zugriff auf die Daten sämtlicher Benutzer. Dies löst AppArmor über die Variable `@HOME` und stellt dem Befehl den Zugriff auf alle Heimatverzeichnisse zur Verfügung.

Schließlich bleibt noch die Qualität der von Novell mitgelieferten Profile zu betrachten. Zunächst enttäuscht die geringe Menge der mitgelieferten Profile. Novell unterstützt in der aktuellen Version 10.1 mit allen Updates nur Profile für die folgenden Applikationen:

- `/bin/netstat`
- `/bin/ping`
- `/lib/ld-2.2.so`
- `/sbin/klogd`

- /sbin/syslogd
- /usr/bin/ldd
- /usr/sbin/identd
- /usr/sbin/mdnsd
- /usr/sbin/nscd
- /usr/sbin/ntpd
- /usr/sbin/traceroute

Alle weiteren Profile in dem Verzeichnis `/etc/apparmor/profiles/extra` gelten als nicht unterstützt. Ein Anwender, der diese Profile nutzen möchte, tut auch gut daran, diese zunächst zu analysieren und zu testen, denn leider enthalten diese Profile teilweise fehlerhafte Pfade oder logische Fehler.

So verweist das Profil `etc.cron.daily.tmpwatch` auf einen Cronjob `tmpwatch` den es auf dieser Version nicht gibt. In dem Profil `usr.lib.RealPlayer10.realplay` wird auf die Applikation `/opt/MozillaFirefox/lib/firefox-bin` verwiesen, die bei SuSE 10.1 unter `/usr/lib/firefox/firefox-bin` installiert wurde. Ähnlich hart-codierte Pfade befinden sich in dem Profil des Acrobat Readers. Ein Upgrade der Java-Virtual-Machine macht das Profil unbrauchbar.

Schließlich kann AppArmor den *Informationsfluss* nicht überwachen. Wenn eine Applikation eine Datei schreibt, kann AppArmor nicht einschränken, welche andere Applikation diese Datei wieder lesen darf, da der verwendete Dateiname beliebig ist und daher nicht von AppArmor als Attribut genutzt werden kann. Dies ist speziell problematisch bei Dateien in Verzeichnissen wie `/tmp`. Hier muss der Administrator den klassischen UNIX-Dateirechten (DAC) vertrauen. AppArmor bietet hier keinen zusätzlichen Schutz.

