

# AppArmor in der Praxis

---

Christian Boltz  
openSUSE community  
cboltz[at]opensuse[dot]org



# Was macht AppArmor?

Die Antwort ist ganz einfach ;-)

- Erlaubt Programmen nur das, was sie eigentlich[tm] machen sollen
- Verhindert alles andere

Ganz so einfach ist es dann doch nicht! ;-)

- erstmal muss AppArmor wissen, was erlaubt ist





# Warum AppArmor?

---

- Ideal wäre sichere, fehlerfreie Software...

# Warum AppArmor?

---

- Ideal wäre sichere, fehlerfreie Software...
- Programmierer können nicht zaubern...



# Warum AppArmor?

- Ideal wäre sichere, fehlerfreie Software...
- Programmierer können nicht zaubern...
- Daher lieber auf die Finger gucken!
  - AppArmor überwacht Programme auf Kernel-Ebene



# Hände hoch! ;-)

- Wer setzt AppArmor (bewusst) ein?
- Wer hat schon Profile über die aa-\* Tools oder YaST angepasst oder erstellt?
- Wer hat schon Profile mit vi / \$EDITOR bearbeitet?
- Gegenprobe: Wer hat noch nicht mit AppArmor gearbeitet?





# Hello world!

---

- Das unvermeidliche Hello World...

```
#!/bin/bash  
echo "Hello World!" > /tmp/hello.txt  
cat /tmp/hello.txt  
rm /tmp/hello.txt
```

- und dafür erstelle ich jetzt ein AppArmor-Profil...

# Hello world!

---

- Das unvermeidliche Hello World...

```
#!/bin/bash  
echo "Hello World!" > /tmp/hello.txt  
cat /tmp/hello.txt  
rm /tmp/hello.txt
```

- **Vorsicht Hacker!**



# Was macht AppArmor?

Überwachung und Beschränkung von

- Dateizugriffen
- Netzwerk-Zugriffen
- Capabilities (z. B. chown, mknod, setuid)
  - man 7 capabilities
- rlimit (aka ulimit)
- Allgemein: Berechtigungen einschränken

Ausnahme:

- “set capability” vergibt zusätzliche Rechte
  - selektives suid-root



# Was macht AppArmor NICHT?

- traditionelle Dateirechte ersetzen
  - “chmod -R 777 /” ist keine gute Idee
- Benutzerrechte ersetzen
  - so wenig wie möglich als root laufen lassen

Speziell für Webserver:

- MySQL Datenbank-Rechte einschränken
  - ein MySQL-Benutzer pro Hosting und Aufgabe
- Benutzer-Input überprüfen
  - Input validieren
  - Input escapen
  - php5-suhosin



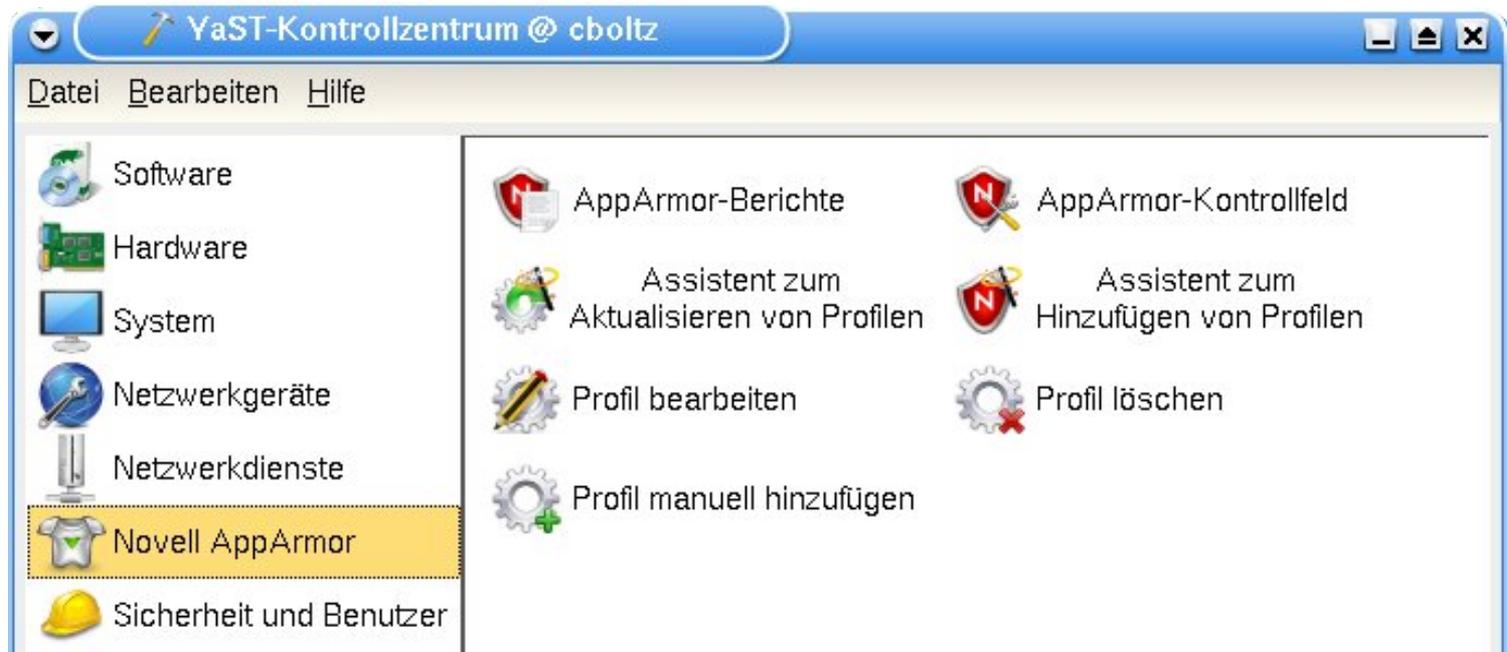


# Ist mein Server jetzt sicher?

---

- Sicherheit besteht aus vielen Einzelteilen
- AppArmor schützt vor vielen (nicht: allen) Exploits
- der Server ist definitiv sicherer als ohne AppArmor ;-)

# YaST AppArmor-Module



- Ich klicke nicht gern auf dem Server herum, daher bevorzugt Kommandozeilen-Tools



# aa-<tab><tab>: Die AppArmor-Tools

---

- aa-unconfined
  - Übersicht, welche Programme geschützt sind
- aa-genprof
  - Neues Profil erstellen
- aa-logprof
  - Vorhandenes Profil ergänzen
- aa-complain
  - Profil in Lernmodus versetzen
  - Verstöße werden geloggt, aber nicht verhindert
- aa-enforce
  - Profil "scharfschalten"
  - nicht erlaubte Aktionen werden geblockt



# aa-unconfined: Status checken

---

```
# aa-unconfined
1552 /usr/lib/postfix/smtpd eingeschränkt
durch '/usr/lib/postfix/smtpd (enforce) '
2879 /usr/sbin/avahi-daemon eingeschränkt
durch '/usr/sbin/avahi-daemon (enforce) '
2955 /usr/sbin/clamd eingeschränkt durch
'/usr/sbin/clamd (enforce) '
3541 /usr/bin/perl (amavisd (master))
eingeschränkt durch '/usr/sbin/amavisd
(complain) '
3839 /usr/sbin/vsftpd nicht eingeschränkt
...
```



# aa-unconfined: Status checken

---

- Grundregel: alle Dienste, die am Internet hängen, sollten geschützt sein

3839 /usr/sbin/vsftpd nicht eingeschränkt

- Dann wird es aber Zeit!



# aa-genprof: Profil erstellen

---

Zwei-Fenster-Taktik:

- im ersten Fenster: aa-genprof /usr/sbin/vsftpd
- im zweiten Fenster mit dem Programm arbeiten

Taktik für die Logauswertung:

- rcvsftpd start / stop
  - erfasst die Basics
  - reduziert die Log-Größe
- das Programm benutzen
- evtl. nach Beenden von genprof das Profil für einige Zeit mit aa-complain in den Lernmodus versetzen
  - besonders bei komplexen Programmen
  - mit aa-logprof “nachlernen”

# Ausführ-Varianten I

---



- inherit (ix)
  - Programm im gleichen Profil ausführen
  - für Hilfsprogramme und Shells (cat, grep, rm, bash)
- child (Cx)
  - Rechtevergabe für “foo aufgerufen von bar”
  - deckt keine eigenständigen Aufrufe von foo ab
  - für Hilfsprogramme, die weniger oder mehr Rechte als das Hauptprogramm haben sollen
- name (Cx -> ...)
  - wie child, wechselt aber zu abstrakten Profilen
  - mehrere Hilfsprogramme können ein gemeinsames abstraktes Profil verwenden

# Ausführ-Varianten II

---



- profile (Px)
  - eigenständiges Profil für Hilfsprogramme
  - gilt auch, wenn das Programm standalone ausgeführt wird
  - keine gute Idee für /bin/bash ;-)
- unconfined (Ux)
  - Hilfsprogramme ohne AppArmor-Überwachung ausführen
  - Praxisbeispiel: sshd schützen, nach dem Login eine uneingeschränkte Shell zur Verfügung stellen
  - nötig für “ps”, weil AppArmor SYS\_PTRACE nie erlaubt (auch wenn es im Profil steht!)
- Umgebung bereinigen?
  - grundsätzlich: ja



# audit.log

---

```
type=APPARMOR_ALLOWED  
msg=audit(1245789190.902:123): [...]
```

- /var/log/audit/audit.log im logdigest aufnehmen
- Timestamp übersetzen:  
date -d @1245789190.902
- APPARMOR\_DENIED – (verhinderte) Verstöße gegen Profile im Erzwingen-Modus
- APPARMOR\_AUDIT – audit-Regeln
- APPARMOR\_ALLOWED – Profile im Lernmodus
- APPARMOR\_HINT ... operation="ptrace" – Fork eines Programms im Lernmodus



# Apache mod\_apparmor

---

- globale Config:  
    AADefaultHatName default\_vhost  
    - ansonsten schlägt AppArmor einen Hat pro Datei (!) vor
- pro VirtualHost:  
    <VirtualHost 1.2.3.4>  
        AADefaultHatName vhost\_irgendwer  
    - ermöglicht die Abschottung verschiedener VirtualHosts
- für einzelne Verzeichnisse:  
    <Directory /some/where>  
        AAHatName irgendwas  
    - besonders empfehlenswert, wenn in einem VirtualHost verschiedene Software (CMS, Forum etc.) zusammen verwendet wird

# Hats?

- Hats sind Unterprofile, zwischen denen ein Programm umschalten kann
- Häufigster Einsatz bei mir: Apache mit einem Hat pro VirtualHost
- Syntax innerhalb der Profile:  

```
^hatname {  
    . . .  
}
```





# mod\_apparmor Basiskonfiguration

---

- /etc/apparmor.d/abstractions/vhost\_cboltz:
  - /home/www/cboltz.de/conf/htpass-webstat r,
  - /home/www/cboltz.de/httpdocs/\*\* r,
  - /home/www/cboltz.de/statistics/logs/access\_log w,
  - /home/www/cboltz.de/statistics/logs/access\_log-20?????? w,
  - /home/www/cboltz.de/statistics/logs/error\_log w,
  - /home/www/cboltz.de/statistics/logs/error\_log-20?????? w,
  - /home/www/cboltz.de/statistics/zugriffe/\* r,
  - /home/www/cboltz.de/tmp/ r,
  - /home/www/cboltz.de/tmp/\*\* rwk,
  - /dev/urandom r,
  - /proc/\*/attr/current w,
  - /usr/share/apache2/error/\*\* r,
  - /usr/share/zoneinfo/ r,



# mod\_apparmor Spezialitäten

---

- abstractions/vhost\_irgendwer automatisch generieren
  - spart gegenüber manueller Profilerstellung pro vHost viel Zeit
- ^HANDLING\_UNTRUSTED\_INPUT macht gern mehr als geplant
  - auf jeden Fall will dieser Hat Schreibzugriff auf die access\_logs und error\_logs aller vHosts
- “Enge” des Profils entscheidet
  - Praxisbeispiel: Forum, das beim Upload von Avatar-Bildern auch \*.php erlaubt...
- “deny owner /\*\*.php rw” als Schutz gegen frisch hochgeladene Exploits
  - blockt aber auch erwünschte Scripte mit owner wwwrun



# AppArmor kreativ

---

- AppArmor als Inventurliste:
  - welcher vHost nutzt welche Scripte/Plugins im serverweiten Verzeichnis?
  - welcher vHost verschickt Mails (Aufruf von sendmail)
  - ...
- AppArmor als Debugging-Hilfe:
  - welche Dateien liest das Programm foo?
  - einfach aa-genprof mitschreiben lassen ;-)
- AppArmor als Lastmonitor
  - “ps Zaux” zeigt, welcher vHost gerade einen Apache-Prozess nutzt / blockiert
- read-only root-Zugang fürs Backup



# Backup: read-only für root

---

Zwei-Komponenten-Lösung:

- SSH-Key in `/root/.ssh/authorized_keys`:  
`command="/root/bin/rsync-shell" ssh-dss 7j1ntgRxts8X...`
- `/root/bin/rsync-shell`:

```
#!/bin/bash
echo "cmd=$SSH_ORIGINAL_COMMAND" |
    logger -t rsync-backup
echo "$SSH_ORIGINAL_COMMAND" |
    grep "^rsync --server --sender" \
    >/dev/null \
    && exec $SSH_ORIGINAL_COMMAND
```



# Backup: read-only für root

- Das zugehörige AppArmor-Profil (leicht gekürzt):

```
/root/bin/rsync-shell {
  #include <abstractions/base>
  #include <abstractions/bash>
  #include <abstractions/consoles>
  #include <abstractions/nameservice>
  capability dac_override,
  capability dac_read_search,
  /bin/bash rix,
  /bin/grep rix,
  /bin/logger Px,
  /root/bin/rsync-shell mr,
  /usr/bin/rsync rix,
  /etc/ r,
  /etc/** r,
  /home/ r,
  /home/** r,
}
```



# Die Zukunft von AppArmor

---

- Upstreaming in den Kernel
- geplante Features für 3.0
  - @`{PID}`
  - @`{UID}`
  - alternative Regel-Syntax `rw /etc/hosts`
    - besser lesbar, macht aber die Pflege von `apparmor.vim` interessanter ;-)
  - `owner=(cb,tux) /tmp/foo* rw`
  - Pfade mit RegEx (statt nur Globbing)
- viele nützliche Kleinigkeiten



# Zum Weiterlesen

---

- <http://en.opensuse.org/AppArmor>
- <http://de.opensuse.org/AppArmor>
  
- openSUSE Security Guide  
<http://www.novell.com/de-de/documentation/opensuse111/>
  
- Mailinglisten: apparmor-general, apparmor-dev
  
- Profil-Repository: <http://apparmor.opensuse.org>
  
- Die Profile meines Servers gibt es als Download unter <http://blog.cboltz.de/plugin/tag/apparmor>

Fragen?

## Lizenz

Diese Präsentation darf entsprechend den Bedingungen der GNU Free Documentation License v 1.3 (<http://www.gnu.org/licenses/fdl.txt>) weitergegeben und verändert werden. Andere Lizenzen sind in Absprache mit dem Autor möglich. Die Lizenzen der verwendeten Fotos können unter den untenstehenden Links eingesehen werden.

## Bildquellen

[www.flickr.com/photos/carbonnyc/2294144289/](http://www.flickr.com/photos/carbonnyc/2294144289/)  
[www.landjugend-rheinhessenpfalz.de/theater-berlin.html](http://www.landjugend-rheinhessenpfalz.de/theater-berlin.html)  
[www.flickr.com/photos/polaroidmemories/2626967595/](http://www.flickr.com/photos/polaroidmemories/2626967595/)  
[www.oldskoolman.de/bilder/technik\\_und\\_bau/werkzeug-baumaterial/axt-klotz/](http://www.oldskoolman.de/bilder/technik_und_bau/werkzeug-baumaterial/axt-klotz/)  
[www.manufactum.de/Produkt/0/1443290/NistkastenWolfgangS.html](http://www.manufactum.de/Produkt/0/1443290/NistkastenWolfgangS.html)  
[www.flickr.com/photos/vrogy/514733529/](http://www.flickr.com/photos/vrogy/514733529/)  
[www.flickr.com/photos/ida-und-bent/248684278/](http://www.flickr.com/photos/ida-und-bent/248684278/)  
[www.flickr.com/photos/kosin-germany/2898566898/](http://www.flickr.com/photos/kosin-germany/2898566898/)  
[www.flickr.com/photos/gotshoo/2336903636/](http://www.flickr.com/photos/gotshoo/2336903636/)

