

# Using Linux VServer

Enrico Scholz

[enrico.scholz@informatik.tu-chemnitz.de](mailto:enrico.scholz@informatik.tu-chemnitz.de)



# Motivation (1)

Einführung

● Motivation (1)

● Motivation (2)

● Anforderungen

● Wünsche

● Lösungsmöglichkeiten

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

Ausblick

## Administrator:

- Vielzahl an Diensten (FTP, HTTP\*, LDAP, KRB, DNS, ...)
- Updates ohne Seiteneffekte
- eigene Hostnamen & IPs
- Abgrenzung der Zugriffsrechte

## Entwickler:

- Test in unterschiedliche Umgebungen (Compiler, Libraries, Programme, Distributionen)
- Bereitstellung von Binaries für verschiedene Distributionen



# Motivation (2)

Einführung

● Motivation (1)

● Motivation (2)

● Anforderungen

● Wünsche

● Lösungsmöglichkeiten

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

Ausblick

## Geschäftsmann:

- gewinnbringendes Vermieten von „root-Servern“ (IP, Rechenleistung, Plattenplatz, eigene Administration)

## Lösung:

- ein physikalischer Rechner pro Server
  - ↪ aber: Hardwarekosten, Raum, Kühlung, USV
- mehrere dedizierte Server auf der selben Hardware
  - ⇒ „virtuelle Server“

# Anforderungen



## Einführung

- Motivation (1)
- Motivation (2)
- Anforderungen
- Wünsche
- Lösungsmöglichkeiten

## vserver

## Sicherheit

## Die Toolsets

## Basisoperationen

## Management

## Ausblick

- Verhalten wie „normaler“ Server (selbe Binaries, selbe Syscalls)
  - Abgrenzung der Prozesse
    - ◆ `kill(2)`, `ptrace(2)`
    - ◆ `/etc/init.d/sshd restart`
  - Abgrenzung des Filesystems
    - ◆ Keine Kollisionen bei Verwendung von Standardpfaden
    - ◆ Wahren von Geheimnissen
  - Keine Hintertüren
    - ◆ direkter Hardwarezugriff (`/dev/hda`)
    - ◆ direkter Kernelzugriff (`/dev/kmem`)
- ⇒ Keine Beeinflussung der Funktion anderer Server oder des Hosts



# Wünsche

## Einführung

- Motivation (1)
- Motivation (2)
- Anforderungen
- Wünsche
- Lösungsmöglichkeiten

## vserver

## Sicherheit

## Die Toolsets

## Basisoperationen

## Management

## Ausblick

- Effizienz
  - ◆ Performance (CPU, I/O)
  - ◆ Speicher (RAM, Platte)
- leichte Managebarkeit
  - ◆ Erstellung
  - ◆ Betrieb
  - ⇒ Nutzung bekannter Mittel
- Limits/Quotas auf Plattenplatz, CPU, Netz
- Migration auf andere physikalische Rechner



# Lösungsmöglichkeiten

## Einführung

---

- Motivation (1)
- Motivation (2)
- Anforderungen
- Wünsche
- Lösungsmöglichkeiten

## vserver

---

## Sicherheit

---

## Die Toolsets

---

## Basisoperationen

---

## Management

---

## Ausblick

---

- spezielle Hardware (S/390)



# Lösungsmöglichkeiten

## Einführung

---

- Motivation (1)
- Motivation (2)
- Anforderungen
- Wünsche
- Lösungsmöglichkeiten

## vserver

---

## Sicherheit

---

## Die Toolsets

---

## Basisoperationen

---

## Management

---

## Ausblick

---

- spezielle Hardware (S/390)
- vmware/bochs/qemu
  - ◆ wie „normaler“ Rechner handhabbar
  - ◆ aber: sehr hoher Ressourcenverbrauch; meist nur für i386



# Lösungsmöglichkeiten

## Einführung

- Motivation (1)
- Motivation (2)
- Anforderungen
- Wünsche
- Lösungsmöglichkeiten

## vserver

## Sicherheit

## Die Toolsets

## Basisoperationen

## Management

## Ausblick

- spezielle Hardware (S/390)
- vmware/bochs/qemu
  - ◆ wie „normaler“ Rechner handhabbar
  - ◆ aber: sehr hoher Ressourcenverbrauch; meist nur für i386
- UML
  - ◆ mittlerer bis hoher Ressourcenverbrauch



# Lösungsmöglichkeiten

## Einführung

- Motivation (1)
- Motivation (2)
- Anforderungen
- Wünsche
- Lösungsmöglichkeiten

## vserver

## Sicherheit

## Die Toolsets

## Basisoperationen

## Management

## Ausblick

- spezielle Hardware (S/390)
- vmware/bochs/qemu
  - ◆ wie „normaler“ Rechner handhabbar
  - ◆ aber: sehr hoher Ressourcenverbrauch; meist nur für i386
- UML
  - ◆ mittlerer bis hoher Ressourcenverbrauch
- SELinux
  - ◆ Anforderungen erfüllbar
  - ◆ keine vollständige Virtualisierung (Hostname, IP)
  - ◆ ???



# Lösungsmöglichkeiten

## Einführung

- Motivation (1)
- Motivation (2)
- Anforderungen
- Wünsche
- Lösungsmöglichkeiten

## vserver

## Sicherheit

## Die Toolsets

## Basisoperationen

## Management

## Ausblick

- spezielle Hardware (S/390)
- vmware/bochs/qemu
  - ◆ wie „normaler“ Rechner handhabbar
  - ◆ aber: sehr hoher Ressourcenverbrauch; meist nur für i386
- UML
  - ◆ mittlerer bis hoher Ressourcenverbrauch
- SELinux
  - ◆ Anforderungen erfüllbar
  - ◆ keine vollständige Virtualisierung (Hostname, IP)
  - ◆ ???
- Linux vserver, BSD Jails, SUN Zones, FreeVPS
  - ◆ Gruppierung von Prozessen
  - ◆ Nutzung gemeinsamer Hardware und Kernel
  - ◆ Neue und bereits existierende Zugriffsbeschränkungen
  - ◆ nahezu kein Overhead



Einführung

---

**vserver**

- Quickstart (1)
- Quickstart (2)
- Eigenschaften (Kernel) (1)
- Eigenschaften (Kernel) (2)
- Userspace
- chroot-Umgebungen

Sicherheit

---

Die Toolsets

---

Basisoperationen

---

Management

---

Ausblick

---

# vserver



# Quickstart (1)

Einführung

vserver

● Quickstart (1)

● Quickstart (2)

● Eigenschaften (Kernel) (1)

● Eigenschaften (Kernel) (2)

● Userspace

● chroot-Umgebungen

Sicherheit

Die Toolsets

Basisoperationen

Management

Ausblick

## 1. Download und Entpacken der Kernel-Quellen:

```
$ wget http://ftp.kernel.org/pub/linux/kernel/v2.x/linux-2.x.y.tar.bz2
$ tar xjf linux-2.x.y.tar.bz2
$ cd linux-2.x.y
```

## 2. Download des entsprechenden vserver-Patches<sup>a</sup> von <http://www.13thfloor.at/vserver/project/> und Anwendung dieses

```
$ bzcat patch-2.x.y-vs1.z.diff.bz2 | patch -p1
```

## 3. Konfiguration, Bau und Installation des Kernels

```
$ make config
$ make dep && make all modules && make install modules_install
```

---

<sup>a</sup> 1.2x – stable, 1.3x und 1.9x – experimentell



# Quickstart (2)

Einführung

vserver

- Quickstart (1)
- Quickstart (2)
- Eigenschaften (Kernel) (1)
- Eigenschaften (Kernel) (2)
- Userspace
- chroot-Umgebungen

Sicherheit

Die Toolsets

Basisoperationen

Management

Ausblick

4. Download der Userspace-Tools<sup>a</sup> (util-vserver) von <http://www.nongnu.org/util-vserver>

5. Konfiguration, Bau und Installation dieser mit

```
$ rpmbuild -ta util-vserver-0.x.y.tar.bz2 \  
  [--without xalan] [--without dietlibc] \  
# rpm -Uvh ...
```

oder

```
$ ./configure [--prefix=...] <options>* && make \  
# make install
```

6. Reboot

<http://linux-vserver.org>

---

<sup>a</sup>  $0.x.y \rightarrow$  stable wenn kein  $y$ , pre bei  $y < 90$ , rc bei  $90 \leq y < 190$  und alpha bei  $190 \leq y$



# Eigenschaften (Kernel) (1)

Einführung

vserver

● Quickstart (1)

● Quickstart (2)

● Eigenschaften (Kernel) (1)

● Eigenschaften (Kernel) (2)

● Userspace

● chroot-Umgebungen

Sicherheit

Die Toolsets

Basisoperationen

Management

Ausblick

- Entwickler: Herbert Pötzl
- ein Multicall-Syscall für gesamte Funktionalität
- Attribute für Prozesse:
  - ◆ numerische Kontext-ID (*xid*)  
→ Prozesse mit  $xid_1$  für  $xid_2$  nicht sichtbar
- Attribute für Prozess-Kontexte:
  - ◆ Hostname bzw. kompletter utsname Eintrag<sup>(2.6)</sup>
  - ◆ system- & kontextspezifische<sup>(2.6)</sup> Capabilities
  - ◆ Flags
  - ◆ Namespace<sup>(2.6)</sup>
  - ◆ Scheduling Parameter<sup>(2.6)</sup>



# Eigenschaften (Kernel) (2)

Einführung

vserver

- Quickstart (1)
- Quickstart (2)
- Eigenschaften (Kernel) (1)
- **Eigenschaften (Kernel) (2)**
- Userspace
- chroot-Umgebungen

Sicherheit

Die Toolsets

Basisoperationen

Management

Ausblick

- Sicherheit größtenteils basierend auf Linux-Capabilities (/usr/include/linux/capability.h), z.B.
  - ◆ kein Erzeugen neuer Devices ohne CAP\_MKNOD
  - ◆ keine Interfacekonfiguration ohne CAP\_NET\_ADMIN
  - ◆ kein Mounten ohne CAP\_SYS\_ADMIN
  - ◆ ...
- Verstecken von Filesystemeinträgen
  - einige Einträge in /proc ohne Capability-Schutz, z.B. /proc/sysrq-trigger oder /proc/scsi/scsi
  - ⇒ Verstecken ausserhalb Hostkontext
- Kontext-Quotas
- ausbruchsichere chroot(2) Umgebung



# Userspace

## Einführung

### vserver

- Quickstart (1)
- Quickstart (2)
- Eigenschaften (Kernel) (1)
- Eigenschaften (Kernel) (2)
- **Userspace**
- chroot-Umgebungen

## Sicherheit

## Die Toolsets

## Basisoperationen

## Management

## Ausblick

- Zwei Toolsets: „vserver“ und „util-vserver“
- Low-level Syscallwrappers
- vserver == chroot-Umgebung + Konfigurationsdaten
- Verwaltung der vserver
  - ◆ Erstellung
  - ◆ Inbetriebnahme/Stoppen
  - ◆ Optimierung
- Konfiguration unter `/etc/vservers/`
- Starten eines Vservers mit “vserver *<id>* start”; Stoppen mit “vserver *<id>* stop”



# chroot-Umgebungen

Einführung

---

vserver

---

- Quickstart (1)
- Quickstart (2)
- Eigenschaften (Kernel) (1)
- Eigenschaften (Kernel) (2)
- Userspace
- chroot-Umgebungen

Sicherheit

---

Die Toolsets

---

Basisoperationen

---

Management

---

Ausblick

---

- gewöhnlich unter `/vservers/<id>`
- normalerweise Dateien und Verzeichnisse wie bei „normalen“ Linuxdistributionen; aber auch Spezialinstallationen möglich
- Distribution im chroot  $\neq$  Host-Distribution



# chroot-Umgebungen

## Einführung

### vserver

- Quickstart (1)
- Quickstart (2)
- Eigenschaften (Kernel) (1)
- Eigenschaften (Kernel) (2)
- Userspace
- chroot-Umgebungen

## Sicherheit

## Die Toolsets

## Basisoperationen

## Management

## Ausblick

- gewöhnlich unter `/vservers/<id>`
  - normalerweise Dateien und Verzeichnisse wie bei „normalen“ Linuxdistributionen; aber auch Spezialinstallationen möglich
  - Distribution im chroot  $\neq$  Host-Distribution
  - chroot-Umgebung grundsätzlich als feindlich anzunehmen:
    - ◆ Ausführen beliebiger Programme als root
    - ◆ Anlegen, Löschen, Umbenennen und Verändern beliebiger Dateien, Symlinks und Verzeichnisse
- ⇒ besondere Sorgfalt und Kernelunterstützung nötig



# Klassische chroot-Attacke

Einführung

---

vserver

---

Sicherheit

● Klassische chroot-Attacke

- Exkurs: Namespaces
- Eindringen in andere chroot's
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

---

Basisoperationen

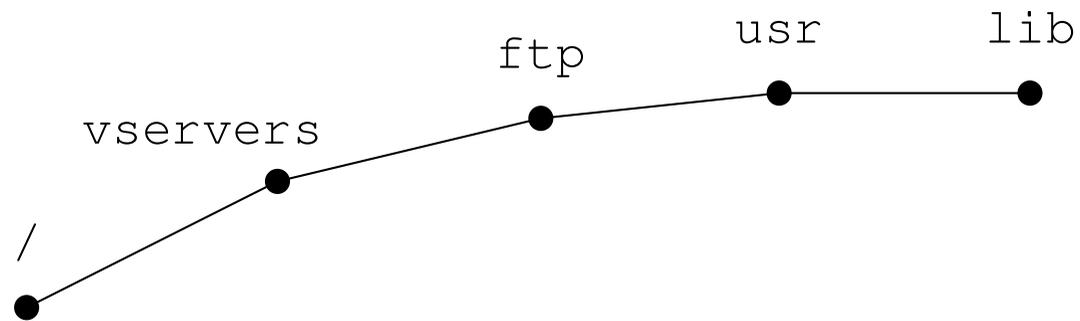
---

Management

---

Ausblick

---





# Klassische chroot-Attacke

Einführung

vserver

Sicherheit

● Klassische chroot-Attacke

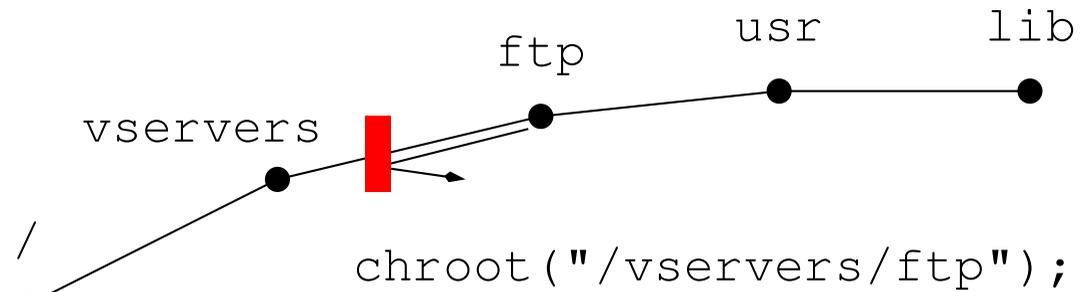
- Exkurs: Namespaces
- Eindringen in andere chroot's
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick



# Klassische chroot-Attacke



Einführung

vserver

Sicherheit

● Klassische chroot-Attacke

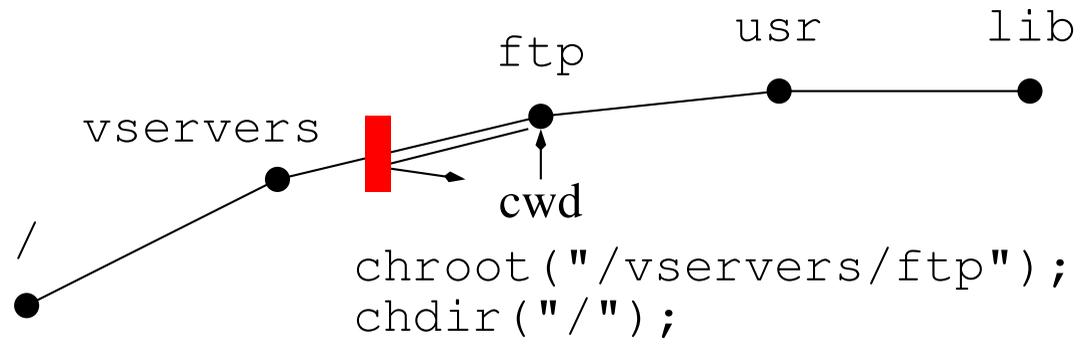
- Exkurs: Namespaces
- Eindringen in andere chroot's
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick



# Klassische chroot-Attacke



Einführung

vserver

Sicherheit

● Klassische chroot-Attacke

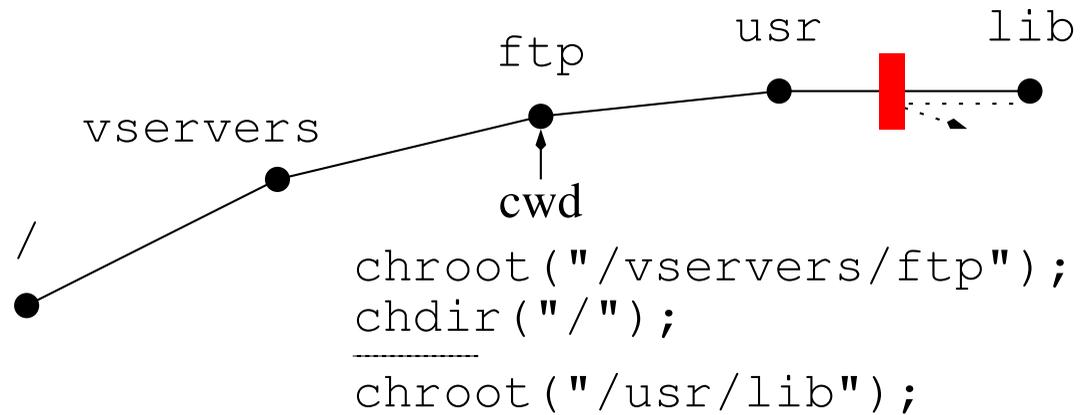
- Exkurs: Namespaces
- Eindringen in andere chroot's
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick



# Klassische chroot-Attacke



Einführung

vserver

Sicherheit

● Klassische chroot-Attacke

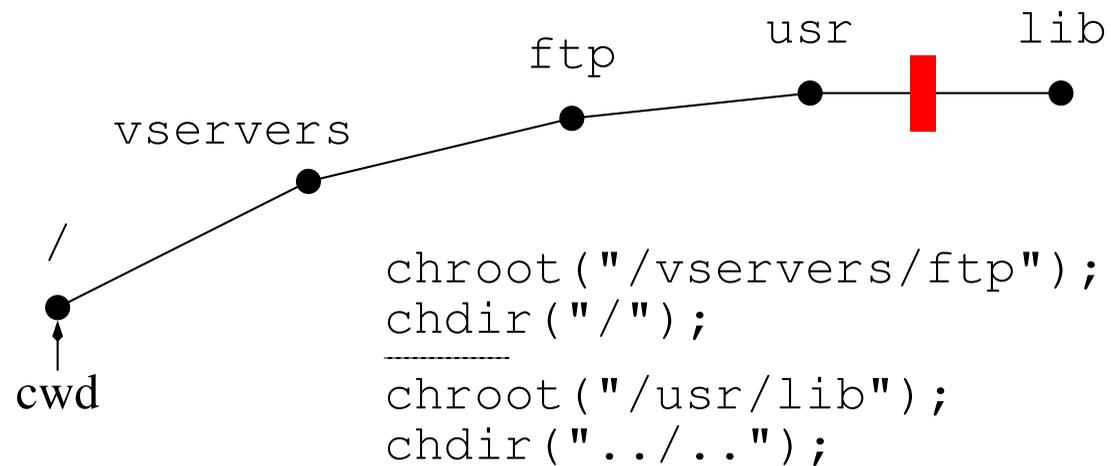
- Exkurs: Namespaces
- Eindringen in andere chroot's
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick





# Klassische chroot-Attacke

Einführung

vserver

Sicherheit

● Klassische chroot-Attacke

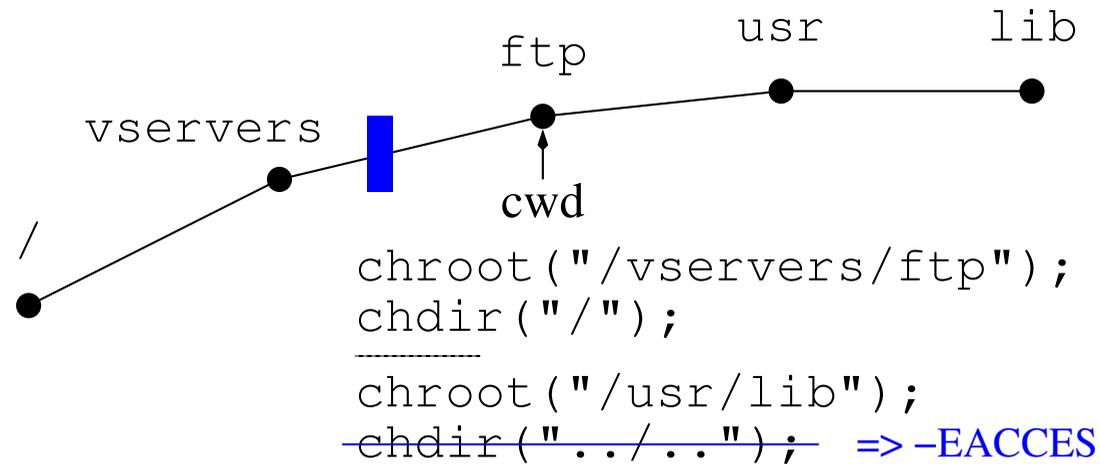
- Exkurs: Namespaces
- Eindringen in andere chroot's
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick



- statische Barriere im Filesystem (Kernelpatch):  
nicht-passierbar durch Prozesse aus nicht-Host Kontexten

# Klassische chroot-Attacke



Einführung

vserver

Sicherheit

● Klassische chroot-Attacke

● Exkurs: Namespaces

● Eindringen in andere chroot's

● Symlink-Attacken (1)

● Symlink-Attacken (2)

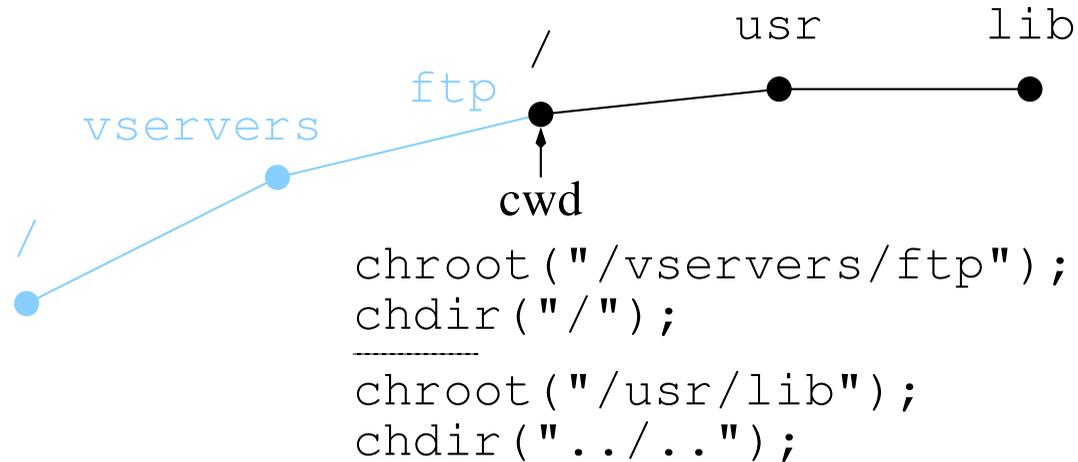
● Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick



## ■ Erzeugung eines neuen “/”:

```
# mount --rbind /vservers/ftp /
```

⇒ Ausführen in separatem Namespace

- ◆ für praktisches Benutzen zusätzliche Features im Kernel nötig (migrate())
- ◆ Aufräumen von /proc/mounts möglich
- ◆ Noch nicht vollständig implementiert



# Exkurs: Namespaces

Einführung

vserver

Sicherheit

● Klassische chroot-Attacke

● Exkurs: Namespaces

● Eindringen in andere chroot's

● Symlink-Attacken (1)

● Symlink-Attacken (2)

● Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick

- Neuer Namespace durch CLONE\_NEWNS; Dokumentation in clone(2) Manpage:

Every process lives in a namespace. The namespace of a process is the data (the set of mounts) describing the file hierarchy as seen by that process.

- relativ neu (Kernel 2.4.19); nicht nur für vserver
- Konflikte mit Automountern

## Beispiel:

```
[root@kosh root]# vnamespace --new sh
sh-2.05b# mount --bind /bin/rm /bin/ls
sh-2.05b# ls /etc/*
... lieber nicht ...
sh-2.05b# exit
[root@kosh root]# ls /etc/*
/etc/DIR_COLORS
...
[root@kosh root]#
```

```
[root@kosh root]# ls /etc/*
/etc/DIR_COLORS
...
[root@kosh root]#
```

# Eindringen in andere chroot's



Einführung

vserver

Sicherheit

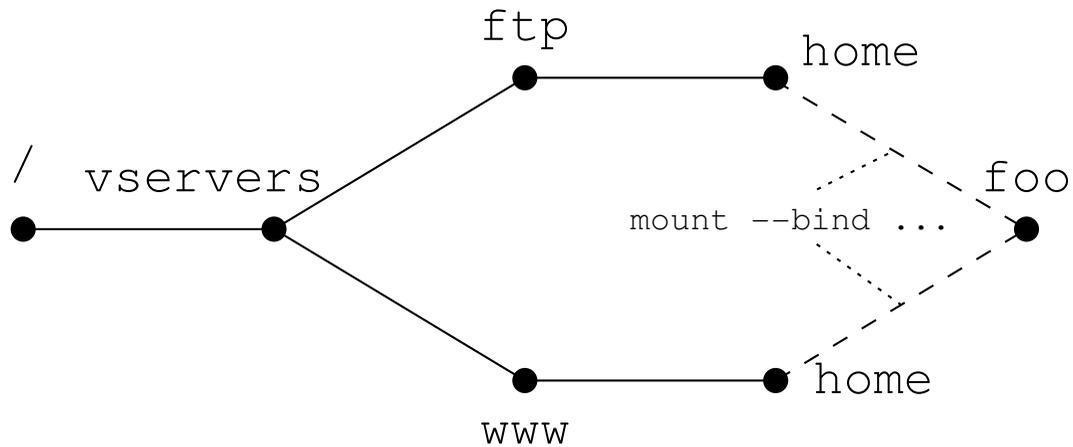
- Klassische chroot-Attacke
- Exkurs: Namespaces
- **Eindringen in andere chroot's**
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick



- Zwei vserver „www“ und „ftp“
- Gemeinsam genutztes /home Verzeichnis

# Eindringen in andere chroot's



Einführung

vserver

Sicherheit

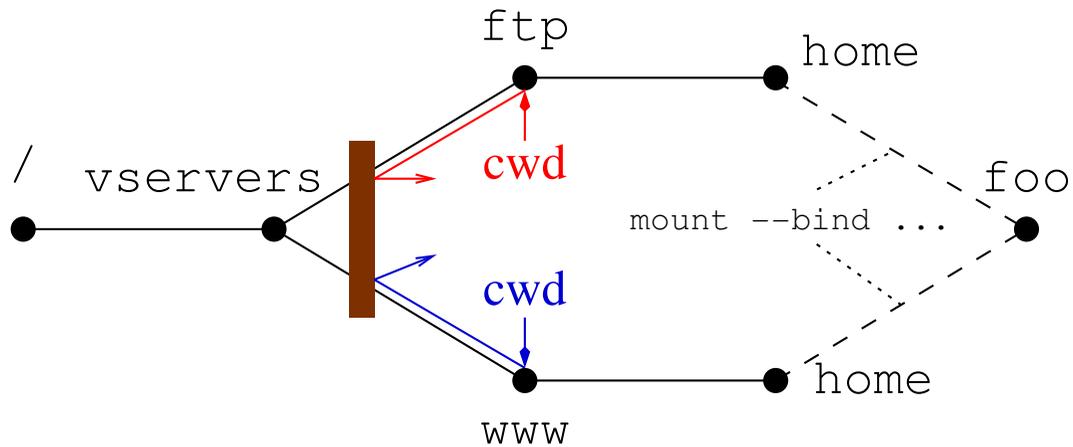
- Klassische chroot-Attacke
- Exkurs: Namespaces
- **Eindringen in andere chroot's**
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick



- Statische Barriere bei /vservers
- root-Rechte für „rot“ in „ftp“; „blau“ nur normaler Nutzer in „www“



# Eindringen in andere chroot's

Einführung

vserver

Sicherheit

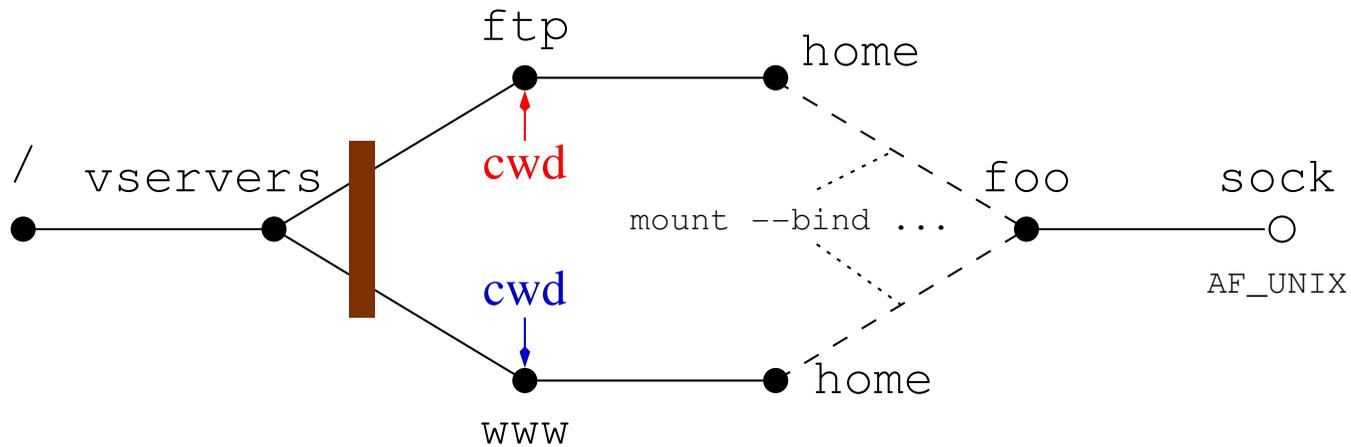
- Klassische chroot-Attacke
- Exkurs: Namespaces
- **Eindringen in andere chroot's**
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick



```
tmp=socket(AF_UNIX,...);  
bind(tmp,"/home/foo/sock");  
listen(tmp)  
s=accept(tmp);
```

```
tmp=socket(AF_UNIX,...)  
connect(s,"/home/foo/sock");
```

# Eindringen in andere chroot's



## Einführung

## vserver

## Sicherheit

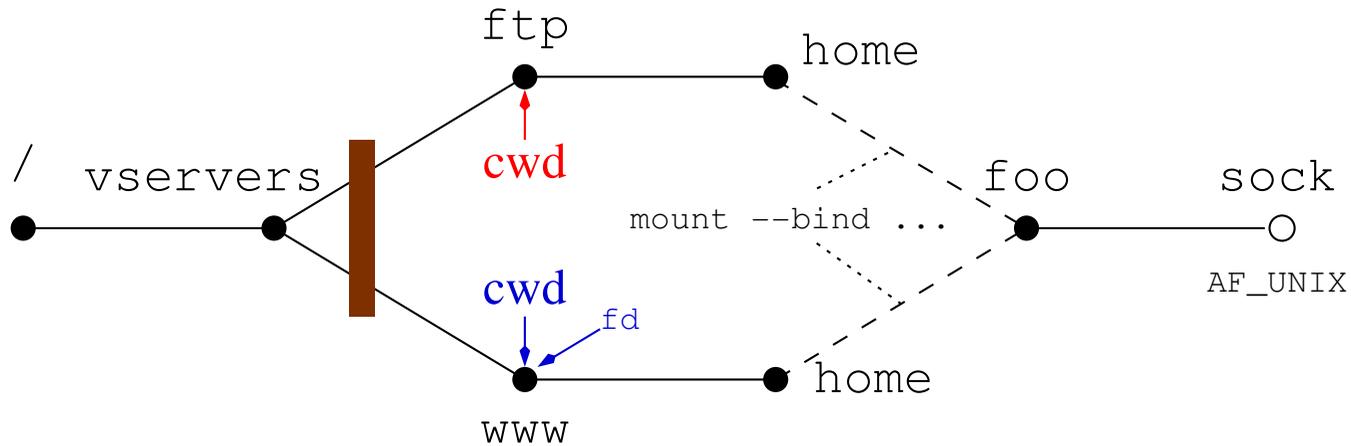
- Klassische chroot-Attacke
- Exkurs: Namespaces
- **Eindringen in andere chroot's**
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

## Die Toolsets

## Basisoperationen

## Management

## Ausblick



```
tmp=socket(AF_UNIX,...);  
bind(tmp, "/home/foo/sock");  
listen(tmp)  
s=accept(tmp);
```

```
tmp=socket(AF_UNIX,...)  
connect(s, "/home/foo/sock");  
fd=open(".", O_RDONLY);
```

# Eindringen in andere chroot's



Einführung

vserver

Sicherheit

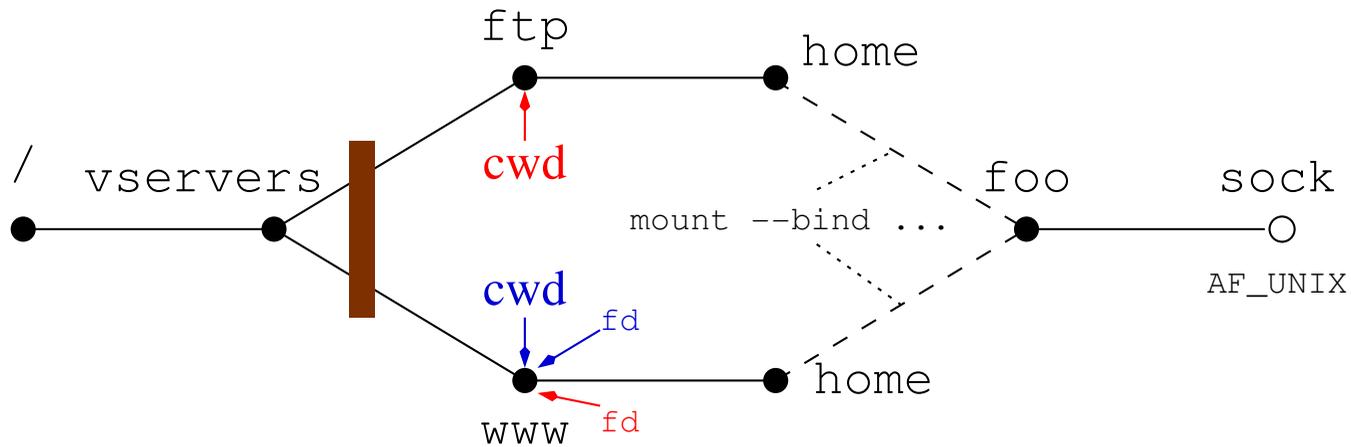
- Klassische chroot-Attacke
- Exkurs: Namespaces
- **Eindringen in andere chroot's**
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick



```
tmp=socket(AF_UNIX,...);  
bind(tmp, "/home/foo/sock");  
listen(tmp)  
s=accept(tmp);  
recvmsg(s, {&fd, SCM_RIGHTS});
```

```
tmp=socket(AF_UNIX,...)  
connect(s, "/home/foo/sock");  
fd=open(".", O_RDONLY);  
sendmsg(s, {fd, SCM_RIGHTS});
```



# Eindringen in andere chroot's

Einführung

vserver

Sicherheit

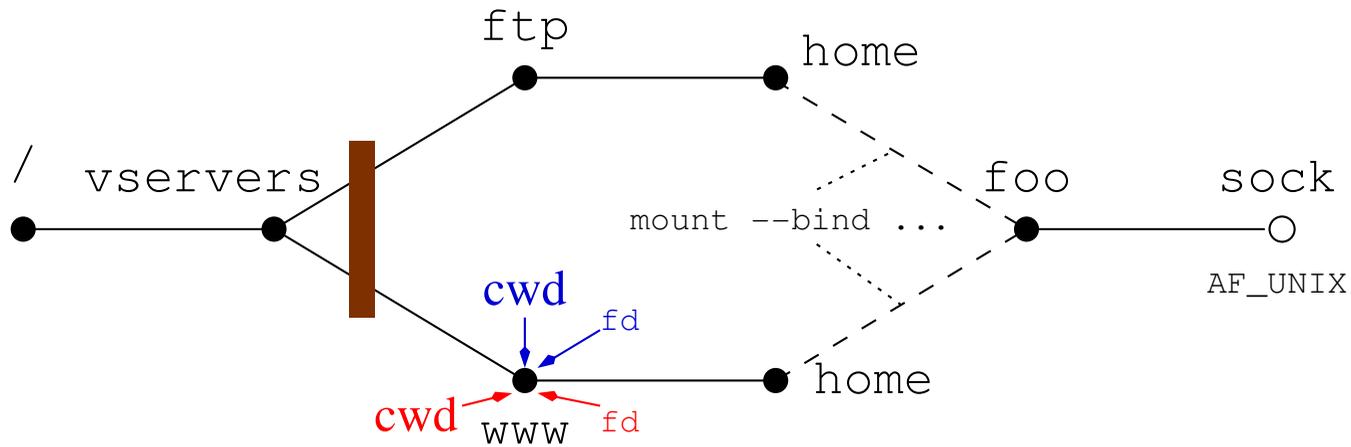
- Klassische chroot-Attacke
- Exkurs: Namespaces
- **Eindringen in andere chroot's**
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick



```
tmp=socket(AF_UNIX,...);  
bind(tmp, "/home/foo/sock");  
listen(tmp)  
s=accept(tmp);  
recvmsg(s, {&fd, SCM_RIGHTS});  
fchdir(fd);
```

```
tmp=socket(AF_UNIX,...)  
connect(s, "/home/foo/sock");  
fd=open(".", O_RDONLY);  
sendmsg(s, {fd, SCM_RIGHTS});
```



# Eindringen in andere chroot's

Einführung

vserver

Sicherheit

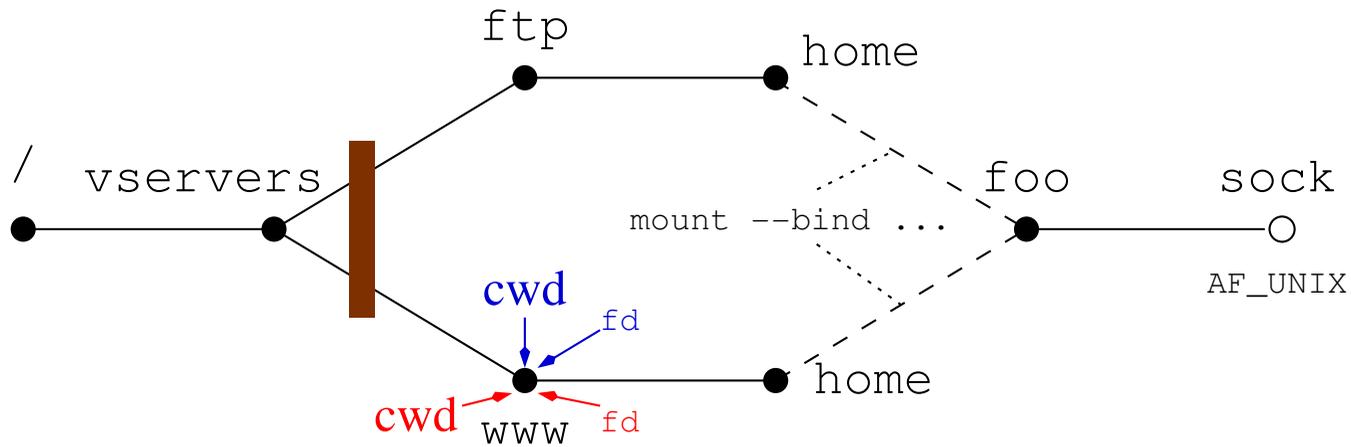
- Klassische chroot-Angriffe
- Exkurs: Namespaces
- **Eindringen in andere chroot's**
- Symlink-Angriffe (1)
- Symlink-Angriffe (2)
- Weitere Angriffe

Die Toolsets

Basisoperationen

Management

Ausblick



```

tmp=socket(AF_UNIX,...);
bind(tmp, "/home/foo/sock");
listen(tmp);
s=accept(tmp);
recvmsg(s, {&fd, SCM_RIGHTS});
fchdir(fd);

```

```

tmp=socket(AF_UNIX,...)
connect(s, "/home/foo/sock");
fd=open(".", O_RDONLY);
sendmsg(s, {fd, SCM_RIGHTS});

```

⇒ root-Rechte für „rot“ in „www“



# Eindringen in andere chroot's

Einführung

vserver

Sicherheit

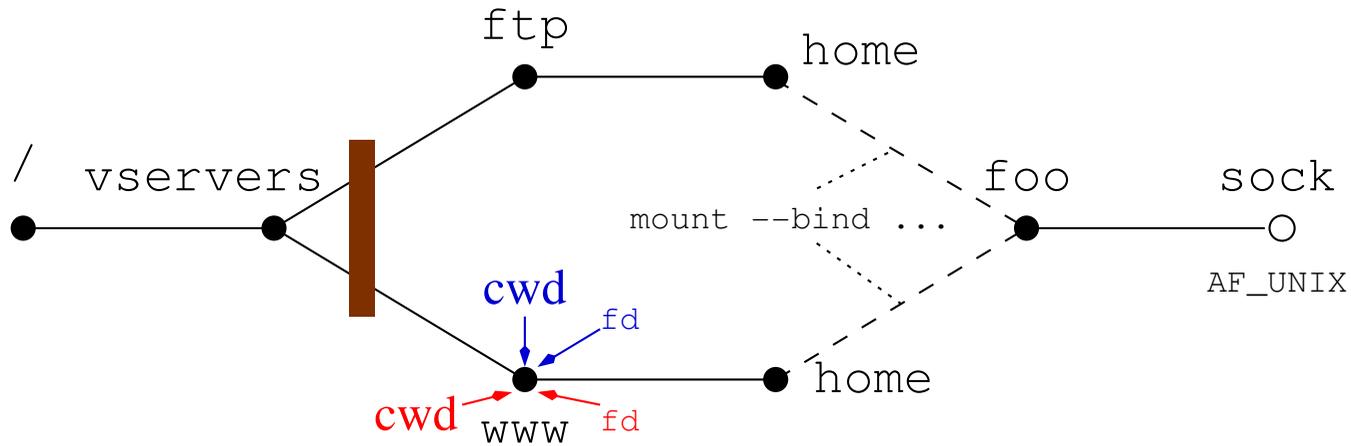
- Klassische chroot-Angriffe
- Exkurs: Namespaces
- **Eindringen in andere chroot's**
- Symlink-Angriffe (1)
- Symlink-Angriffe (2)
- Weitere Angriffe

Die Toolsets

Basisoperationen

Management

Ausblick



```
tmp=socket(AF_UNIX,...);
bind(tmp, "/home/ftp/sock");
listen(tmp);
s=accept(tmp);
recvmsg(s, {&fd, SCM_RIGHTS});
fchdir(fd);
open("etc/passwd",...);
```

```
tmp=socket(AF_UNIX,...)
connect(s, "/home/ftp/sock");
fd=open(".", O_RDONLY);
sendmsg(s, {fd, SCM_RIGHTS});
```

⇒ root-Rechte für „rot“ in „www“

- Ungelöst in vserver; evtl. verhinderbar mittels SELinux oder teilweise durch Namespaces



# Symlink-Attacken (1)

Einführung

---

vserver

---

Sicherheit

---

- Klassische chroot-Attacke
- Exkurs: Namespaces
- Eindringen in andere chroot's
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

---

Basisoperationen

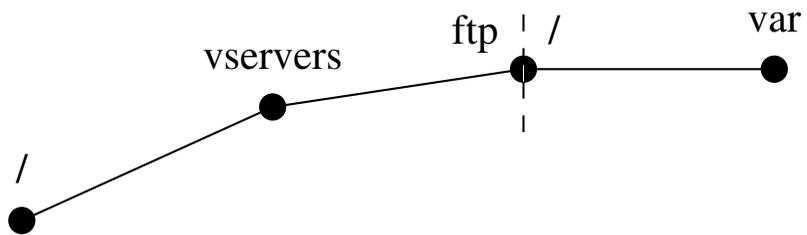
---

Management

---

Ausblick

---





# Symlink-Attacken (1)

Einführung

vserver

Sicherheit

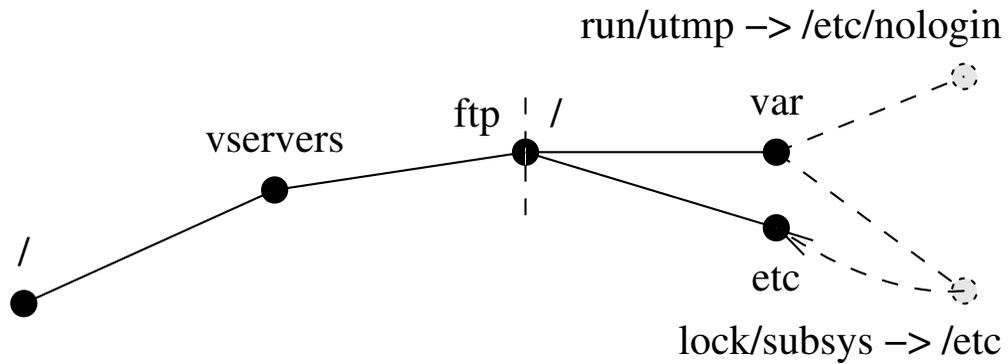
- Klassische chroot-Attacke
- Exkurs: Namespaces
- Eindringen in andere chroot's
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick



Vserver-Admin:

```
# ln -s /etc /var/lock/subsys
```

```
# ln -s /etc/nologin /var/run/utmp
```

# Symlink-Attacken (1)



Einführung

vserver

Sicherheit

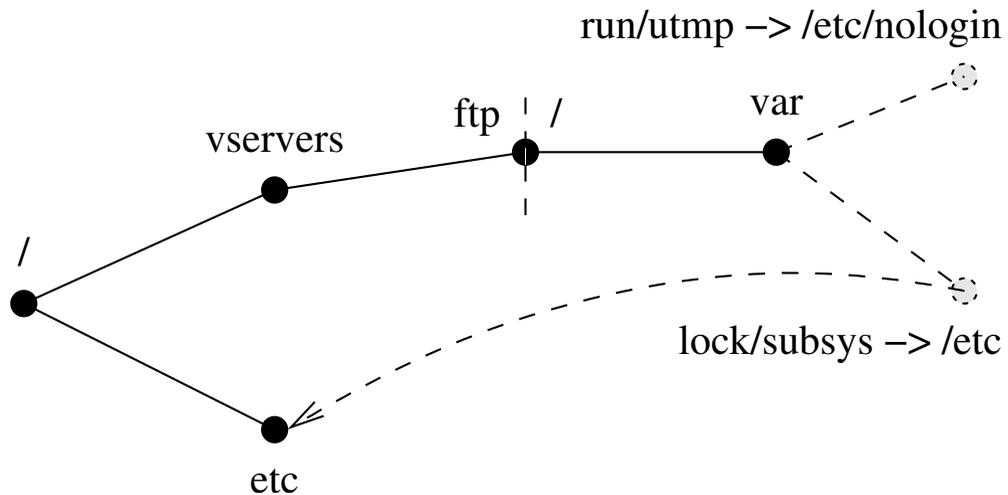
- Klassische chroot-Attacke
- Exkurs: Namespaces
- Eindringen in andere chroot's
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick



Vserver-Admin:

```
# ln -s /etc /var/lock/subsys
```

```
# ln -s /etc/nologin /var/run/utmp
```

**Aktionen, ausgeführt vom Host-Admin in „/“ :**

```
# rm -f /vserver/ftp/var/lock/subsys/*
```

```
# touch /vservers/ftp/var/run/utmp
```

```
# mount /dev/hda1 /vserver/ftp/var/lock/subsys
```



# Symlink-Attacken (2)

Einführung

vserver

Sicherheit

- Klassische chroot-Attacke
- Exkurs: Namespaces
- Eindringen in andere chroot's
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick

- Verhinderung von Symlinkangriffen durch *sicheres* Betreten der Verzeichnisse; meist implementiert durch

```
chroot(vserver_rootdir);
chdir(destination_directory);
action();
```

- Durchführen von Operationen nur in “.” (exec-cd Tool)

```
// Usage: exec-cd <dir> <cmd> <args>*
old_fd = open("/", O_RDONLY);
chroot(".");
chdir(argv[1]);
new_fd = open(".", O_RDONLY);
fchdir(old_fd);
chroot(".");
fchdir(new_fd);
execv(argv[2], argv+2);
```

- z.B.:

```
# cd /vservers/ftp && exec-cd /var/lock/subsys mount /dev/hda1 ''
```



# Weitere Attacken

Einführung

vserver

Sicherheit

- Klassische chroot-Attacke
- Exkurs: Namespaces
- Eindringen in andere chroot's
- Symlink-Attacken (1)
- Symlink-Attacken (2)
- Weitere Attacken

Die Toolsets

Basisoperationen

Management

Ausblick

- Modifikation von Dateien, die normalerweise nur durch root modifizierbar (/etc/passwd, rpm-Datenbank)
  - ⇒ Provokation von Overflows
  - ⇒ Ausführen von Code im Host-Kontext
  - Lösung:** Halten wichtiger Dateien außerhalb des Vservers; Hilfsprogramme im Vserver-Kontext
- Dynamisches Nachladen von Bibliotheken (/lib/libnss\_\*) (auch funktionale Einschränkungen)
  - Lösung:** dietlibc statt glibc
- Races beim Bewegen im Filesystem
  - Lösung:** sicheres Verzeichniswechseln; Erzwingen gestoppter Vserver
- kein chroot(2) vor Kontext-Betreten
  - ⇒ Hijacken via ptrace(2)
  - Lösung:** nicht tun. . .



# util-vserver, stable (1)

Einführung

vserver

Sicherheit

Die Toolsets

● util-vserver, stable (1)

● util-vserver, stable (2)

● util-vserver, alpha

● Konfiguration

Basisoperationen

Management

Ausblick

- „vserver“ und stable-Branch von „util-vserver“:
  - ◆ nahezu identische Funktionalität
  - ◆ Fork von „util-vserver“ bei „vserver 0.23“
- weit verbreitet
- sehr gute Dokumentation



# util-vserver, stable (1)

Einführung

vserver

Sicherheit

Die Toolsets

● util-vserver, stable (1)

● util-vserver, stable (2)

● util-vserver, alpha

● Konfiguration

Basisoperationen

Management

Ausblick

- „vserver“ und stable-Branch von „util-vserver“:
    - ◆ nahezu identische Funktionalität
    - ◆ Fork von „util-vserver“ bei „vserver 0.23“
  - weit verbreitet
  - sehr gute Dokumentation
  - viele offene Wünsche
  - nicht einsetzbar in feindlichen Umgebungen aufgrund vielfältiger Möglichkeiten für Symlinkattacken
- ⇒ komplettes Redesign erforderlich
- keine Weiterentwicklung
  - keine Unterstützung neuer Features



# util-vserver, stable (2)

Einführung

vserver

Sicherheit

Die Toolsets

● util-vserver, stable (1)

● **util-vserver, stable (2)**

● util-vserver, alpha

● Konfiguration

Basisoperationen

Management

Ausblick

## Konfiguration in /etc/vservers/<id>.conf

```
IPROOT="192.168.5.32 192.168.5.64"  
IPROOTDEV=eth0  
S_HOSTNAME=ftp.nowhe.re  
ONBOOT=yes  
S_DOMAINNAME=  
S_NICE=5  
S_FLAGS="lock nproc fakeinit"  
ULIMIT="-HS -u 200"  
S_CAPS=""
```

- bash-Skriptlet; eingebunden mit source
- evtl. weiteres Skript /etc/vservers/<id>.sh für Aufgaben nach/vor dem Starten/Stoppen



# util-vserver, alpha

Einführung

vserver

Sicherheit

Die Toolsets

● util-vserver, stable (1)

● util-vserver, stable (2)

● util-vserver, alpha

● Konfiguration

Basisoperationen

Management

Ausblick

- Designziele:
  - ◆ leicht erweiterbar
  - ◆ keine Races; resistent gegen Symlinkangriffe
  - ◆ Eingebaute Lösungen für Standardaufgaben
  - ◆ Unterstützung neuer Kernelfeatures
- Beibehaltung der Basiskommandos von stable, aber viele neue Befehle und Reimplementierung alter
- Neues Konfigurationsschema
  - ◆ parsebar durch C und shell
  - ◆ managebar durch cfengine
  - ◆ Unterstützung neuer Features



# util-vserver, alpha

Einführung

vserver

Sicherheit

Die Toolsets

● util-vserver, stable (1)

● util-vserver, stable (2)

● util-vserver, alpha

● Konfiguration

Basisoperationen

Management

Ausblick

- Designziele:
  - ◆ leicht erweiterbar
  - ◆ keine Races; resistent gegen Symlinkangriffe
  - ◆ Eingebaute Lösungen für Standardaufgaben
  - ◆ Unterstützung neuer Kernelfeatures
- Beibehaltung der Basiskommandos von stable, aber viele neue Befehle und Reimplementierung alter
- Neues Konfigurationsschema
  - ◆ parsebar durch C und shell
  - ◆ managebar durch cfengine
  - ◆ Unterstützung neuer Features
- kaum Dokumentation

<http://www.linux-vserver.org/index.php?page=alpha+util-vserver>

# Konfiguration

Einführung

vserver

Sicherheit

Die Toolsets

● util-vserver, stable (1)

● util-vserver, stable (2)

● util-vserver, alpha

● **Konfiguration**

Basisoperationen

Management

Ausblick

## ■ Konfiguration in `/etc/vservers/<id>/` Verzeichnis

```
/etc/vservers/ftp
|-- capabilities
|-- context
|-- flags
|-- fstab
|-- interfaces
|   |-- 00
|   |   |-- ip
|   |   '-- name
|   |-- bcast
|   |-- dev
|   '-- mask
|-- run -> /var/run/vservers/ftp
|-- run.rev -> ../.defaults/run.rev
'-- vdir -> /etc/vservers/.defaults/vdirbase/ftp
```

- Dateien & Symlinks; meist Ein-Eintrag-pro-Zeile/Datei
- Identifikation eines vservers durch Pfad des Konfigurationsverzeichnisses; chroot-Pfad frei wählbar
- nur formelle Dokumentation



# vcontext (1)

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

● vcontext (1)

● vcontext (2)

● vattribute

● chbind

● Verschiedenes (1)

● Verschiedenes (2)

Management

Ausblick

- früher: chcontext, aber keine Unterstützung neuer Technologien
- Erzeugung (`--create`) und Betreten (`--migrate`) von Prozess-Kontexten
- Beim Betreten auf Sicherheit achten! (`ptrace(2)`)
- Zwischen Erzeugung und Betreten gewöhnlich noch andere Operationen
- Aufruf gewöhnlich als:

```
vcontext --create -- \  
vattribute --set -- \  
vlimit ... -- \  
vsched ... -- \  
vcontext --migrate-self --endsetup -- \  
<command>
```

# vcontext (2)



Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

● vcontext (1)

● vcontext (2)

● vattribute

● chbind

● Verschiedenes (1)

● Verschiedenes (2)

Management

Ausblick

## Beispiel:

```
# vcontext --create ps axh
New security context is 49153
5440 pts/1      R      0:00 ps axh

# vcontext --migrate --xid 43 ps axh
5068 ?          S      0:00 /sbin/syslogd
5102 ?          S      0:00 /usr/sbin/exim4 -bd -q30m
5108 ?          S      0:00 /usr/sbin/inetd
5112 ?          S      0:00 /usr/sbin/atd
5115 ?          S      0:00 /usr/sbin/cron
5447 pts/1      R      0:00 ps axh
```



Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

● vcontext (1)

● vcontext (2)

● **vattribute**

● chbind

● Verschiedenes (1)

● Verschiedenes (2)

Management

Ausblick

- Setzen/Löschen von Attributen und Capabilities
- Darstellungsmöglichkeiten dieser Werte:
  - ◆ Zeichenkette
  - ◆ Zahl: Interpretation als Bitmuster
  - ◆ Prefix ‘~’ oder ‘!’: Löschen des entsprechenden Bitmusters
  - ◆ Prefix ‘^’: Interpretation als Bitnummer

## Beispiel:

```
# vcontext --create vattribute --set --flag hidemount cat /proc/mounts
# vcontext --create -- \
    vattribute --set --secure -- \
    vcontext --endsetup --migrate-self -- \
    mknod /tmp/test c 1 2
New security context is 49183
mknod: '/tmp/test': Operation not permitted
```



# chbind

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

● vcontext (1)

● vcontext (2)

● vattribute

● **chbind**

● Verschiedenes (1)

● Verschiedenes (2)

Management

Ausblick

- Binden von IPs an Prozesse
- Zukunft ungewiss... evtl. vollkommen anderes Networking oder Ablösung durch vnet

## Beispiel:

```
# chbind --ip 10.1.0.1 cat /proc/self/status | grep ipv4root
ipv4root is now 10.1.0.1
ipv4root: 0100010a/00ffffff
ipv4root_bcast: ffffffff
ipv4root_refcnt: 2
```



# Verschiedenes (1)

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

● vcontext (1)

● vcontext (2)

● vattribute

● chbind

● **Verschiedenes (1)**

● Verschiedenes (2)

Management

Ausblick

## vkill

- Atomares Senden von Signalen an Prozess-Kontext

## vnnamespace

- Erzeugen und Betreten von Namespaces

## vlimit

- Setzen & Anzeigen von Beschränkungen



# Verschiedenes (2)

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

● vcontext (1)

● vcontext (2)

● vattribute

● chbind

● Verschiedenes (1)

● Verschiedenes (2)

Management

Ausblick

## vuname

- Anzeigen und Ändern von utsname-Einträgen

## vserver-info

- Abfrage einzelner Attribute von Kontexten und vservern
- wichtig für Bugreports:  
vserver-info – SYSINFO



Einführung

---

vserver

---

Sicherheit

---

Die Toolsets

---

Basisoperationen

---

**Management**

- vserver Erzeugung
- vserver ... build
- vserver ... start
- vserver ... stop
- vserver ... enter|exec
- vps
- vserver-stat
- vrpm (1)
- vrpm (2)
- vapt-get
- setattr, showattr (1)
- setattr, showattr (2)
- setattr, showattr (3)
- vunify

Ausblick

---

# Management



# vserver Erzeugung

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

● vserver Erzeugung

● vserver ... build

● vserver ... start

● vserver ... stop

● vserver ... enter|exec

● vps

● vserver-stat

● vrpm (1)

● vrpm (2)

● vapt-get

● setattr, showattr (1)

● setattr, showattr (2)

● setattr, showattr (3)

● vunify

Ausblick

■ keine „normale“ Systeminstallation möglich

■ BSD Jails:

```
# make -C /usr/src DESTDIR=/vservers/foo install
```



# vserver Erzeugung

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

● vserver Erzeugung

● vserver ...build

● vserver ...start

● vserver ...stop

● vserver ...enter|exec

● vps

● vserver-stat

● vrpm (1)

● vrpm (2)

● vapt-get

● setattr, showattr (1)

● setattr, showattr (2)

● setattr, showattr (3)

● vunify

Ausblick

- keine „normale“ Systeminstallation möglich

- BSD Jails:

```
# make -C /usr/src DESTDIR=/vservers/foo install
```

- Fedora Core:

```
# make -C /usr/src DESTDIR=/vservers/foo install
make: Entering directory '/usr/src'
make: *** No rule to make target 'install'. Stop.
make: Leaving directory '/usr/src'
```

- Vielzahl an Distributionen mit unterschiedlichen Installationsmöglichkeiten



# vserver Erzeugung

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

● vserver Erzeugung

● vserver ... build

● vserver ... start

● vserver ... stop

● vserver ... enter|exec

● vps

● vserver-stat

● vrpm (1)

● vrpm (2)

● vapt-get

● setattr, showattr (1)

● setattr, showattr (2)

● setattr, showattr (3)

● vunify

Ausblick

- keine „normale“ Systeminstallation möglich

- BSD Jails:

```
# make -C /usr/src DESTDIR=/vservers/foo install
```

- Fedora Core:

```
# make -C /usr/src DESTDIR=/vservers/foo install
make: Entering directory '/usr/src'
make: *** No rule to make target 'install'. Stop.
make: Leaving directory '/usr/src'
```

- Vielzahl an Distributionen mit unterschiedlichen Installationsmöglichkeiten

⇒ Implementierung *einiger* in util-vserver:

- ◆ „apt-rpm“ für Fedora/RH vserver
- ◆ „debootstrap“ für Debian vserver
- ◆ „skeleton“ für Basis Verzeichnisstruktur + Konfiguration



# vserver ... build

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

● vserver Erzeugung

● vserver ... build

● vserver ... start

● vserver ... stop

● vserver ... enter|exec

● vps

● vserver-stat

● vrpm (1)

● vrpm (2)

● vapt-get

● setattr, showattr (1)

● setattr, showattr (2)

● setattr, showattr (3)

● vunify

Ausblick

## ■ Dokumentation durch „vserver - build --help“

## ■ Beispiel:

```
◆ # vserver test0 build -m apt-rpm --hostname test0.nowhe.re \  
  --interface 10.0.1.0 --netdev eth0 --netprefix 23 \  
  --context 42 -- -d fcl
```

```
◆ # vserver test1 build -m debootstrap --hostname test1.nowhe.re \  
  --interface 10.0.1.1 --netdev eth0 --netprefix 23 \  
  --context 43 -- -d sarge
```

```
◆ # vserver test2 build -m skeleton --hostname test2.nowhe.re \  
  --interface 10.0.1.2 --netdev eth0 --netprefix 23 \  
  --context 44
```

## ■ Konfiguration der Parameter (Mirror, Paketlisten) durch /etc/vservers/.defaults/apps/debootstrap/\* und /etc/vservers/.distributions/\*



# vserver ... start

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

● vserver Erzeugung

● vserver ... build

● vserver ... start

● vserver ... stop

● vserver ... enter|exec

● vps

● vserver-stat

● vrpm (1)

● vrpm (2)

● vapt-get

● setattr, showattr (1)

● setattr, showattr (2)

● setattr, showattr (3)

● vunify

Ausblick

1. Erzeugung eines Namespaces
2. Erzeugung von Netzwerkinterfaces
3. Mounten von Verzeichnissen
4. Erzeugung von Prozess- und Netzwerkkontext
5. Aktivierung von Beschränkungen (Capabilities, Limits)
6. Aufruf des init-Prozesses:
  - Shortcut über „/etc/rc.d/rc 3“, oder
  - reguläres /sbin/init – oft Menge unerwünschter Aktionen⇒ `fakeinit` Mechanismus nötig (`getpid()==1`)

**Achtung:** mindestens ein laufender Prozess benötigt

## Beispiel:

```
# vserver test0 start
# vserver --debug test1 start
```



# vserver ... stop

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

● vserver Erzeugung

● vserver ... build

● vserver ... start

● **vserver ... stop**

● vserver ... enter|exec

● vps

● vserver-stat

● vrpm (1)

● vrpm (2)

● vapt-get

● setattr, showattr (1)

● setattr, showattr (2)

● setattr, showattr (3)

● vunify

Ausblick

- entweder
  - ◆ Senden von `SIGINT` an init-Prozess, oder
  - ◆ Ausführen von „`/etc/rc.d/rc 6`“
- explizites „`vkill -xid <xid> -s 9`“
- Rückgängigmachen von „`vserver ... start`“
- bei Verwendung von Namespaces kein explizites Unmounten nötig

## Beispiel:

```
# vserver test0 stop
# vserver --debug test1 stop
```



# vserver ... enter|exec

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

● vserver Erzeugung

● vserver ... build

● vserver ... start

● vserver ... stop

● vserver ... enter|exec

● vps

● vserver-stat

● vrpm (1)

● vrpm (2)

● vapt-get

● setattr, showattr (1)

● setattr, showattr (2)

● setattr, showattr (3)

● vunify

Ausblick

- Ausführen von Kommandos innerhalb des vservers
  - Ähnliche Aktionen wie bei „vserver ... start“, aber Betreten statt Erzeugen von Namespaces und Kontexten
- ⇒ kein Überschreiben eventueller Parameter und Restriktionen
- möglichst nur für Administrationsaufgaben, nicht für regulären Betrieb (z.B. fehlende /dev/pts Einträge)

## Beispiel:

```
# vserver test0 exec ps axh
 640 ?          S          0:00 syslogd -m 0
1188 pts/1      R          0:00 ps axh
# vserver test1 enter
test1:/#

# uname -a
Linux delenn 2.6.5ensc-0.3 #1 Thu Apr 15 ... 2004 i686 i686 i386 GNU/Linux
# vserver test1 exec uname -a
SCO UnixWare test1.nowhe.re 7.1 #1 Sat Feb 29 ... 2003 s390 GNU/Linux
```



# vps

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

● vserver Erzeugung

● vserver ...build

● vserver ...start

● vserver ...stop

● vserver ...enter|exec

● **vps**

● vserver-stat

● vrpm (1)

● vrpm (2)

● vapt-get

● setattr, showattr (1)

● setattr, showattr (2)

● setattr, showattr (3)

● vunity

Ausblick

- Anzeige aller Prozesse auf dem Host
- Ausführung von ps in speziellem Watcher-Kontext (XID 1)

## Beispiel:

```
# vps ax
  PID CONTEXT          TTY      STAT     TIME COMMAND
    1      0 MAIN                ?        S        0:05 /sbin/mini
    2      0 MAIN                ?        SWN      0:00 [ksoftirqd/0]
    ...
 5068    43 test1             ?        S        0:00 /sbin/syslogd
 5102    43 test1             ?        S        0:00 /usr/sbin/exim4 -bd -q30m
 5108    43 test1             ?        S        0:00 /usr/sbin/inetd
 5112    43 test1             ?        S        0:00 /usr/sbin/atd
 5115    43 test1             ?        S        0:00 /usr/sbin/cron
    ...
 5256    42 test0             ?        S        0:00 syslogd -m 0
    ...
 5276      1 ALL_PROC          pts/1    S        0:00 vps ax
 5277      1 ALL_PROC          pts/1    R        0:00 ps ax
```



# vserver-stat

## ■ Überblick über laufende vserver bzw. Prozess-Kontexte

### Beispiel:

```
# vserver-stat
CTX    PROC    VSZ      RSS    userTIME    sysTIME    UPTIME    NAME
0      37    43.8M    4.6K    0m37s86    0m22s52    13h00m44    root server
42     1     1.5M    145    0m00s00    0m00s00    2m56s60    test0
43     5    10.5M    965    0m00s10    0m00s00    8m53s31    test1
```

```
# vserver-stat
CTX    PROC    VSZ      RSS    userTIME    sysTIME    UPTIME    NAME
0      48    143.9M    3.5K    19h09m59    8h40m24    56d45h05    root server
2      3     6.7M    172    3h01m34    1h22m00    28d29h14    vpn
82     17    90.6M    784    4h44m09    2h13m18    28d26h53    cvs
133    5     1.6M    52     31m03s55    5m46s86    23d33h26    paris
146    11    48.9M    2K     37m07s12    8m55s74    28d26h53    mirror
147    7     3.8M    180    10h46m17    2h48m54    28d21h46    mirror-master
153    9     74.6M    4K     36m31s24    7m08s33    28d25h46    ldap1
```

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

- vserver Erzeugung
- vserver ... build
- vserver ... start
- vserver ... stop
- vserver ... enter|exec
- vps
- vserver-stat
- vrpm (1)
- vrpm (2)
- vapt-get
- setattr, showattr (1)
- setattr, showattr (2)
- setattr, showattr (3)
- unify

Ausblick



# vrpm (1)

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

- vserver Erzeugung
- vserver ... build
- vserver ... start
- vserver ... stop
- vserver ... enter|exec
- vps
- vserver-stat
- vrpm (1)
- vrpm (2)
- vapt-get
- setattr, showattr (1)
- setattr, showattr (2)
- setattr, showattr (3)
- unify

Ausblick

- externes oder internes Halten der Datenbank
- Vorteil intern: rpm Funktionalität innerhalb des vservers
- Vorteil extern: leichtes Bootstrappen („vserver ... build“)
- Wechsel zwischen beiden Methoden via  
vserver ... pkgmgmt externalize|internalize

## Syntax:

```
vrpm <vserver>+ -- <rpm-options>+
```

## Internes vrpm:

- Realisiert über „vserver ... exec rpm“



# vrpm (2)

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

- vserver Erzeugung
- vserver ... build
- vserver ... start
- vserver ... stop
- vserver ... enter|exec
- vps
- vserver-stat
- vrpm (1)
- vrpm (2)
- vapt-get
- setattr, showattr (1)
- setattr, showattr (2)
- setattr, showattr (3)
- vunify

Ausblick

## Externes vrpm:

- LD\_PRELOAD Wrapper für `execv(3)`, `getpwnam(3)` & Co.
  - ⇒ Ausführung von %scriptlets im Kontext des vservers
  - ⇒ NSS Lookups beim Entpacken der Pakete
- kompliziertes Mounten der Datenbank in vserver, so dass Zugriff durch vserver-Prozesse oder %scriptlets unmöglich
- Daten unter `/etc/vservers/<id>/apps/pkgmgmt/...` bzw. `/vservers/.pkg/<id>/rpm`

## Beispiel:

```
# vrpm test0 -- -q glibc fedora-release rpm
glibc-2.3.2-101.4
fedora-release-1-3
package rpm is not installed
# vrpm test0 -- -Uvh /tmp/tetex-2.0.2-13.i386.rpm
```



# vapt-get

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

- vserver Erzeugung
- vserver ... build
- vserver ... start
- vserver ... stop
- vserver ... enter|exec
- vps
- vserver-stat
- vrpm (1)
- vrpm (2)
- vapt-get
- setattr, showattr (1)
- setattr, showattr (2)
- setattr, showattr (3)
- vunify

Ausblick

- für RPM basierende vserver: externes und internes Management möglich
- sonst realisiert über „vserver ... exec apt-get“

## Syntax:

```
vapt-get <vserver>+ -- <apt-get-options>+
```

## Beispiel:

```
# vapt-get test0 -- install bzip2-libs
...
Preparing... ##### [100%]
  1:bzip2-libs ##### [100%]
Done.

# vapt-get test1 -- install libbz2-1.0
...
Unpacking libbz2-1.0 (from ../libbz2-1.0_1.0.2-1_i386.deb) ...
Setting up libbz2-1.0 (1.0.2-1) ...
```



# setattr, showattr (1)

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

- vserver Erzeugung
- vserver ... build
- vserver ... start
- vserver ... stop
- vserver ... enter|exec
- vps
- vserver-stat
- vrpm (1)
- vrpm (2)
- vapt-get
- setattr, showattr (1)
- setattr, showattr (2)
- setattr, showattr (3)
- vunify

Ausblick

- meist: Menge von vservern mit gleicher Distribution
  - ⇒ Installation und Ausführung identischer Pakete, Binaries, Daten
- Idee: Kopieren via Hardlinks („In A B“)
  - ⇒ Einsparung von Plattenplatz
  - ⇒ Einsparung von Speicher (Mapping von Programmen und Libraries)



# setattr, showattr (1)

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

- vserver Erzeugung
- vserver ... build
- vserver ... start
- vserver ... stop
- vserver ... enter|exec
- vps
- vserver-stat
- vrpm (1)
- vrpm (2)
- vapt-get
- setattr, showattr (1)
- setattr, showattr (2)
- setattr, showattr (3)
- vunify

Ausblick

- meist: Menge von vservern mit gleicher Distribution
  - ⇒ Installation und Ausführung identischer Pakete, Binaries, Daten
- Idee: Kopieren via Hardlinks („In A B“)
  - ⇒ Einsparung von Plattenplatz
  - ⇒ Einsparung von Speicher (Mapping von Programmen und Libraries)
  - ↔ Manipulationen möglich, da Änderungen auf jedem vserver sichtbar:
    - echo mycode >/usr/sbin/httpd**
  - ◆ Kein COW oder unionfs unter Linux



# setattr, showattr (2)

- Lösung: spezielles immutable-Flag; z.B. „chattr +i ...“
  - ◆ nicht setzbar ausserhalb von Hostkontext
- ↪ Paketmanagement (Updates) nicht mehr möglich

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

- vserver Erzeugung
- vserver ... build
- vserver ... start
- vserver ... stop
- vserver ... enter|exec
- vps
- vserver-stat
- vrpm (1)
- vrpm (2)
- vapt-get
- setattr, showattr (1)
- setattr, showattr (2)
- setattr, showattr (3)
- vunify

Ausblick



# setattr, showattr (2)

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

- vserver Erzeugung
- vserver ... build
- vserver ... start
- vserver ... stop
- vserver ... enter|exec
- vps
- vserver-stat
- vrpm (1)
- vrpm (2)
- vapt-get
- setattr, showattr (1)
- setattr, showattr (2)
- setattr, showattr (3)
- unify

Ausblick

- Lösung: spezielles immutable-Flag; z.B. „chattr +i ...“
    - ◆ nicht setzbar ausserhalb von Hostkontext
    - ↪ Paketmanagement (Updates) nicht mehr möglich
  - weiteres Flag, so dass:
    - ◆ Verhinderung von Modifikationen
    - ◆ Erlauben von Löschen
  - low-level Funktionalität in setattr und showattr Tools
    - ◆ Änderung der Modifizierbarkeit mit „-iunlink“
    - ◆ Änderung der Sichtbarkeit
    - ◆ Setzen des chroot-Barriere Flags
- ⇒ „setattr --help“

# setattr, showattr (3)



Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

- vserver Erzeugung
- vserver ... build
- vserver ... start
- vserver ... stop
- vserver ... enter|exec
- vps
- vserver-stat
- vrpm (1)
- vrpm (2)
- vapt-get
- setattr, showattr (1)
- setattr, showattr (2)
- setattr, showattr (3)
- unify

Ausblick

## Beispiel:

```
# touch /vservers/test0/{a,b,c}
# ln /vservers/test0/{a,b,c} /vservers/test1/
# setattr --iunlink /vservers/test0/a
# chattr +i /vservers/test0/b
# showattr /vservers/test0/{a,b,c}
---bUI- /vservers/test0/a
---buI- /vservers/test0/b
---bui- /vservers/test0/c

# vserver test0 enter
[root@test0]# echo a>a
bash: a: Permission denied
[root@test0]# echo a>b
bash: a: Permission denied
[root@test0]# echo a>c
[root@test0]#

[root@test0]# rm -f a b c
rm: cannot remove 'b': Operation not permitted

# vserver test1 enter
test1:/# cat /c
a
```



# vunify

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

- vserver Erzeugung
- vserver ... build
- vserver ... start
- vserver ... stop
- vserver ... enter|exec
- vps
- vserver-stat
- vrpm (1)
- vrpm (2)
- vapt-get
- setattr, showattr (1)
- setattr, showattr (2)
- setattr, showattr (3)
- **vunify**

Ausblick

- Fortsetzung des setattr-Konzepts, aber auf ganze Verzeichnisbäume
- Funktion:
  1. Finden gleicher Dateien
  2. Setzen des `iunlink` Flags
  3. Erzeugung eines Hardlinks
- Verwendung statischer Exclude-Listen, und Informationen des Paketmanagements über Konfigurationsdateien



Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

- vserver Erzeugung
- vserver ... build
- vserver ... start
- vserver ... stop
- vserver ... enter|exec
- vps
- vserver-stat
- vrpm (1)
- vrpm (2)
- vapt-get
- setattr, showattr (1)
- setattr, showattr (2)
- setattr, showattr (3)
- **vunify**

Ausblick

- Fortsetzung des setattr-Konzepts, aber auf ganze Verzeichnisbäume
- Funktion:
  1. Finden gleicher Dateien
  2. Setzen des `iunlink` Flags
  3. Erzeugung eines Hardlinks
- Verwendung statischer Exclude-Listen, und Informationen des Paketmanagements über Konfigurationsdateien
- vollständige Fedora Core 1 Installation nur ca. 30 MB nicht teilbarer Dateien
  - ⇒ 2.6 GB Plattenplatz für 20 vserver á 2 GB



# Ausblick

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

Ausblick

● **Ausblick**

● Referenzen

● Fragen?

- neues Netzwerkkonzept: Tagging von Paketen, iptables, Routingtabellen
- Dokumentation
- Testsuits
- alpha → beta → stable (vor GNU Hurd??)



# Referenzen

Einführung

vserver

Sicherheit

Die Toolsets

Basisoperationen

Management

Ausblick

● Ausblick

● **Referenzen**

● Fragen?

- Projekthomepage <http://linux-vserver.org>
- util-vserver <http://www.nongnu.org/util-vserver>
- #vserver auf [oftc.net](http://oftc.net)



**Fragen?**