„Nicht alle Dateien sind, was sie zu seien scheinen!“

Mythos Symlink

Ein Vortrag von

Marius Schwarz

„Was genau sind Symlinks?“

Da stellen wir uns einmal ganz dumm,

... und fangen am Anfang an.

„Was genau sind Symlinks?“

„Symlink“ ist eine verkürzte Sprachweise für „Symbolic-Link“

„Was genau sind Symlinks?“

Auf Deutsch würde ein Symlink mit „Verweis“ übersetzt werden,

besser und bekannter ist „Verknüpfung“.

„Was genau sind Symlinks?“

Ein Symlink ist eine Verbindung von zwei oder mehr Stellen im Filesystem

„Was genau sind Symlinks?“

Beispiel:

```
[marius@eve test]$ ll
insgesamt 28
-rw-rw-r--. 1 marius marius 11562  7. Aug 12:49 beispiel.txt.bz2
lrwxrwxrwx. 1 marius marius    16  8. Aug 12:47 Dies-hier-auch.gustav -> beispiel.txt.bz2
lrwxrwxrwx. 1 marius marius    16  8. Aug 12:46 Dies-ist-nur-ein-Link -> beispiel.txt.bz2
-rwx------. 1 marius marius   159  7. Aug 13:38 test1.sh
-rwxr-xr-x. 1 marius marius   287  7. Aug 12:50 test.sh
```

Wie man sehen kann, nimmt ein Symlink nur 16 Bytes ein.
Die reguläre Datei, auf die er verlinkt/zeigt, ist aber 11.562 Bytes groß.

„Was genau sind Symlinks?“

Im Gegensatz zu einer regulären Datei,

nimmt ein Symlink kaum Platz auf der Festplatte ein.

„Wozu braucht man Symlinks?“

Symlinks werden gebraucht,

wenn der Inhalt einer Datei an zwei oder mehr Stellen im Filesystem benötigt wird.

„Wozu braucht man Symlinks?“

Der Vorteil ist, daß, auch wenn man den Symlink öffnet,
man eigentlich die reguläre Datei aufmacht.

Beispiel:

```
[marius@eve test]$ md5sum beispiel.txt.bz2
5473ff5dd4567ab0057fd3360a8b4b7a  beispiel.txt.bz2
[marius@eve test]$ md5sum Dies-ist-nur-ein-Link
5473ff5dd4567ab0057fd3360a8b4b7a  Dies-ist-nur-ein-Link
```

„Wozu braucht man Symlinks?“

Löscht man einen Symlink,

wird die verlinkte Datei ***nicht*** gelöscht.

„Wozu braucht man Symlinks?“

Wozu kann man das noch einsetzen?

„Wozu braucht man Symlinks?“

UNIX Systeme wurden in den 1960er Jahren entwickelt.

„Wozu braucht man Symlinks?“

UNIX Systeme wurden in den 1960er Jahren entwickelt.

In den letzten 60 Jahren, gab es die eine oder andere Neuprogrammierung bestehender Software. Daraus resultiert das Problem, daß sich Dateisystemstrukturen geändert haben, also Dateien an anderen „Orten“ liegen, als früher.

„Wozu braucht man Symlinks?“

UNIX Systeme wurden in den 1960er Jahren entwickelt.

In den letzten 60 Jahren, gab es die eine oder andere Neuprogrammierung bestehender Software. Daraus resultiert das Problem, daß sich Dateisystemstrukturen geändert haben, also Dateien an anderen „Orten“ liegen, als früher.

Um eine Rückwärtskompatibilität zu gewährleisten,
kann man durch Symlinks alte Strukturen nachbilden,
ohne Daten doppelt auf der Platte haben zu müssen.

„Wozu braucht man Symlinks?“

Beispiel: Linux

Seit Anbeginn der Linuxzeitrechnung, wurden Programme unter /bin/ abgespeichert.

„Wozu braucht man Symlinks?“

Beispiel: Linux

Seit Anbeginn der Linuxzeitrechnung, wurden Programme unter /bin/ abgespeichert.

Heutzutage werden diese Programme aber in /usr/bin/ gespeichert.

„Wozu braucht man Symlinks?“

Beispiel: Linux

Seit Anbeginn der Linuxzeitrechnung, wurden Programme unter /bin/ abgespeichert.

Heutzutage werden diese Programme aber in /usr/bin/ gespeichert.

Scripte, deren Inhalt noch auf /bin/ referenziert, müssen aber weiter funktionieren.

„Wozu braucht man Symlinks?“

Lösung:

```
[marius]$ ls -la /bin
lrwxrwxrwx. 1 root root 7 13. Jul 2018  /bin -> /usr/bin
```

/bin verlinkt einfach auf das neue Ziel /usr/bin . Alte Script laufen damit einfach weiter, als wenn sich nie etwas geändert hätte.

„Gibt es noch andere Links?“

Hardlinks:

Hardlinks sind anders als Symlinks, Verknüpfungen auf INodes.

Hardlinks unterscheiden sich darin, daß Sie nur innerhalb einer logischen Festplattenpartition möglich sind und nicht über verschiedene Dateisysteme gehen können.

„Gibt es noch andere Links?“

Hardlinks:

Löscht man das referenzierte Ziel eines Symlinks, sind alle Symlinks tod.

Löscht man das referenzierte Ziel eines Hardlinks, passiert erstmal gar nichts.
Der Inhalt der Datei „verschwindet“ erst, wenn der letzte Hardlink gelöscht wird.

„Wie setzt man einen SymLink“

Der Befehl dazu lautet: „ln“

Generell:

„ln -s ZIEL linkname“

Beispiel:

„ln -s /usr/bin /bin“

womit /bin dann auf → /usr/bin zeigt.

„Spaß mit Links“

These: „Mit Symlinks lassen sich Webserver hacken!“

„(Un)Spaß mit Links“

JA, LEIDER :(

„(Un)Spaß mit Links“

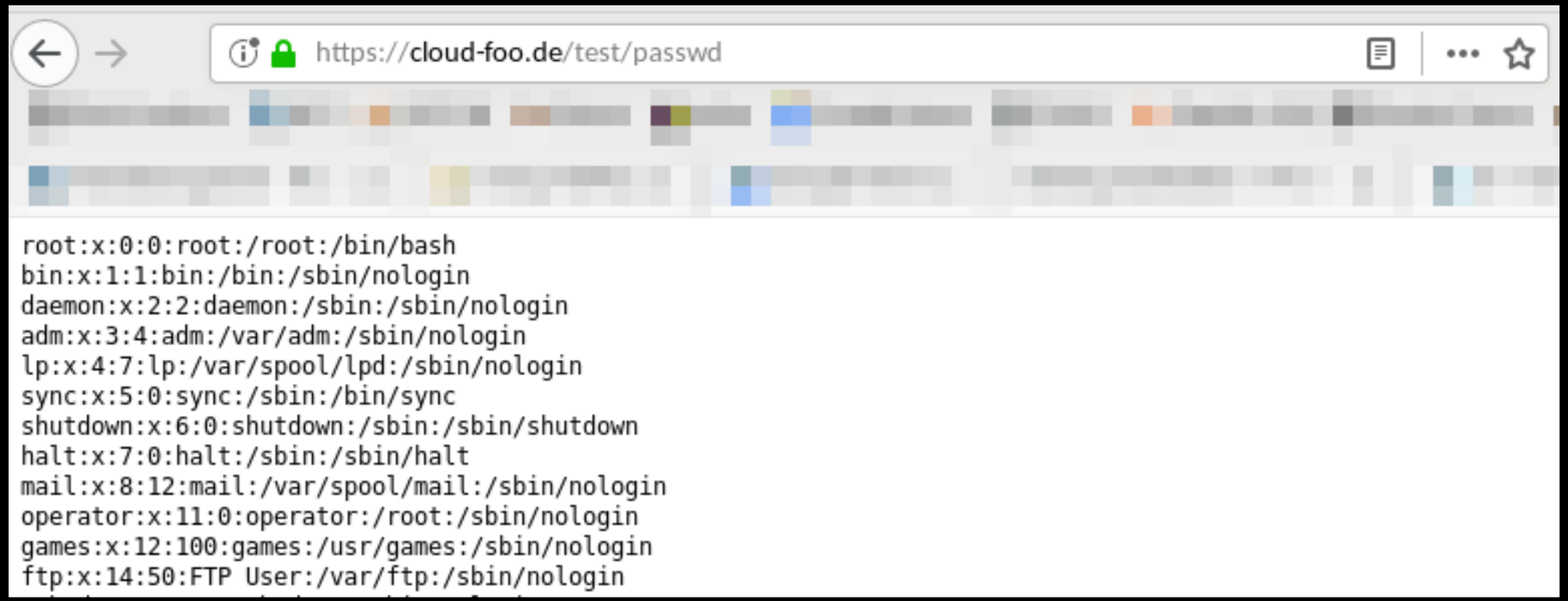
Beispiel:

Apache Webserver mit „Options +FollowSymLinks“ aka „Default“

```
[cloud-foode test]$ ls -la passwd
lrwxrwxrwx 1 cloud-foode cloud-foode   11  8. Aug 13:39 passwd -> /etc/passwd
[cloud-foode test]$
```

„(Un)Spaß mit Links“

Ergebnis:



„(Un)Spaß mit Links“

Lösung:

Apache im Vhost so einstellen:

„Options -FollowSymLinks +FollowSymLinksifOwnerMatch“

Dann kann ein Link nur noch benutzt werden,
wenn Quelle und Ziel dem gleichen User gehören.

Da Dateien in /etc/ normalerweise root gehören,
kann der Symlink nicht mehr funktionieren.

Hardlinks kann ohnehin nur Root anlegen.

„(Un)Spaß mit Links“

Hintergrund:

Man macht sich hier zunutze,
daß der Apache Webserver deutlich mehr Rechte hat,
Dateien im System zu lesen, als der Benutzer selbst,
denn letztlich ist es der Apache, der den Symlink öffnet.

Wäre das nicht so, könnte der Apache seinen Job nicht machen.

„(Un)Spaß mit Links“

Backups mit TAR:

Erstellt man mit TAR ein Backup des Systems,
kommen darin zwangsweise Symlinks vor,
da diese auf fast jedem System zum Einsatz kommen:

Beispiel: /etc/alternatives

„(Un)Spaß mit Links“

Backups mit TAR:

Packt man so ein TAR Backup wieder aus und bricht es „auf der Hälfte des Weges“ wieder ab, weil die Verzeichnisse, die man wollte schon ausgepackt sind, ist es möglich, daß die Symlinks „fehlen“.

Tar restauriert diese erst, wenn es komplett beendet wurde:

https://marius.bloggt-in-braunschweig.de/2019/08/02/tar-und-die-symlinks/

Demnächst in diesem Kino: „Spaß mit Flaggen“

ENDE.