Socket-Details für Linux Admins

Carsten Grohmann

7. September 2011

Agenda

- Netzwerkverbindungen eines Prozesses anzeigen
- Socket-Optionen anzeigen
- Weiterführende Informationen
- Nachwort

wget ist meistens vorhanden ...

```
# wget http://ftp.gwdg.de/pub/linux/knoppix/KNOPPIX_V6.4.4CD-2011-01-30-DE.iso
# wget ftp://ftp.gwdg.de/pub/linux/knoppix/KNOPPIX_V6.4.4CD-2011-01-30-DE.iso
```

Details zu allen Netzwerkverbindungen anzeigen

```
# netstat
Aktive Internetverbindungen (ohne Server)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 noname:46184           ftp.gwdg.de:http        VERBUNDEN
[...]
```

Details zur Verbindung mit der GWDG anzeigen

```
# lsof | grep gwdg
wget      2491    carsten    3u     IPv4     244604      0t0      TCP↘
      noname:46184->ftp.gwdg.de:http (ESTABLISHED)
```

```
# lsof -c wget
COMMAND   PID     USER   FD   TYPE DEVICE SIZE/OFF       NODE NAME
wget     2491  carsten  cwd    DIR    8,7     8192  235030264 /home/↘
      carsten
wget     2491  carsten  rtd    DIR    8,6     4096        128 /
wget     2491  carsten  txt    REG    8,6   368572  168119513 /usr/bin/↘
      wget
wget     2491  carsten  mem    REG    8,6    71488   33580832 /lib/i386-↘
      linux-gnu/i686/cmov/libresolv-2.13.so
[...]
wget     2491  carsten    3u  IPv4 244604      0t0       TCP noname↘
      :46184->ftp.gwdg.de:http (ESTABLISHED)
wget     2491  carsten    4w   REG    8,7 13594272 234881164 /home/↘
      carsten/KNOPPIX_V6.4.4CD-2011-01-30-DE.iso


# lsof -i
COMMAND    PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
wget      2491  carsten    3u  IPv4 244604      0t0  TCP noname:46184->↘
      ftp.gwdg.de:http (ESTABLISHED)
```

Protokoll, Host und Service filtern: lsof -i

Alle TCP-Verbindungen

```
# lsof -iTCP
COMMAND   PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
wget     2491 carsten    3u  IPv4 244604      0t0  TCP noname:46184->↘
      ftp.gwdg.de:http (ESTABLISHED)
```

Alle Verbindungen mit dem Service ftp (Port 21 - Kontrollkanal)

```
# lsof -i:ftp
COMMAND  PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
wget    7523 carsten    3u  IPv4 303781      0t0  TCP noname:42178->↘
      ftp.gwdg.de:ftp (ESTABLISHED)
```

Alle Verbindungen mit dem Server ftp.gwdg.de

```
# lsof -i@ftp.gwdg.de
COMMAND  PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
wget    2580 carsten    3u  IPv4 244604      0t0  TCP noname:46257->↘
      ftp.gwdg.de:http (ESTABLISHED)
```

Syntax lsof -i

`[46][Protokoll][@Hostname|IP-Adresse][:Service|Port]`

- 46 - IPv4 oder IPv6
- `Protokoll` - `TCP` oder `UDP`
- ...

Nur spezifizierte Elemente werden gefiltert

Protokollstatus filtern: lsof -s

Alle TCP-Verbindungen im Status LISTEN oder CLOSE_WAIT

```
# lsof -iTCP -sTCP:LISTEN,CLOSE_WAIT
COMMAND   PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
lighttpd 2704 www-data    4u  IPv4  10119      0t0  TCP *:http (↘
    LISTEN)
master   2869     root   12u  IPv4  10241      0t0  TCP localhost:↘
    smtp (LISTEN)
```

Alle aufgebauten TCP-Verbindungen (Status ESTABLISHED)

```
# lsof -i -sTCP:ESTABLISHED
COMMAND  PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
wget    5012 carsten    3u  IPv4  28492      0t0  TCP noname:44203->↘
    ftp.gwdg.de:http (ESTABLISHED)
```

Alle TCP-Verbindungen die nicht im Status CLOSE_WAIT sind

```
# lsof -itcp -stcp:^close_wait
COMMAND   PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
lighttpd 2704 www-data    4u  IPv4  10119      0t0  TCP *:http (↘
    LISTEN)
master   2869     root   12u  IPv4  10241      0t0  TCP localhost:↘
    smtp (LISTEN)
wget     5020   carsten    3u  IPv4  32134      0t0  TCP noname↘
    :44205->ftp.gwdg.de:http (ESTABLISHED)
```

Syntax lsof -s <Protokoll:Status>

```
-s Protokoll:[^]Status
```

- `Protokoll` - `TCP` oder `UDP`
- `Status` - Verbindungsstatus z.B. `LISTEN`, `ESTABLISHED`, `IDLE`
- Groß- und Kleinschreibung wird ignoriert
- `^` negiert den Status
- `-i` und `-s` sollten zusammen verwendet werden
- Bei `-s` ohne Argumente zeigt `lsof` immer die Größe der offenen Dateien an

Weitere nützliche Optionen für lsof

- `-n` Deaktiviert die Auflösung von IP-Adressen in Hostnamen
- `-P` Deaktiviert die Auflösung von Port-Nummern in Service-Namen
- ...

pfiles unter Solaris

```
# pfiles 1661
1661:   /usr/lib/sendmail -bd -q15m
  Current rlimit: 1024 file descriptors
   0: S_IFCHR mode:0666 dev:284,0 ino:6815752 uid:0 gid:3 rdev:13,2
      O_RDONLY|O_LARGEFILE
      /devices/pseudo/mm@0:null
[...]
   5: S_IFCHR mode:0000 dev:284,0 ino:47600 uid:0 gid:0 rdev:21,45
      O_WRONLY FD_CLOEXEC
      /devices/pseudo/log@0:conslog
   6: S_IFSOCK mode:0666 dev:291,0 ino:11580 uid:0 gid:0 size:0
      O_RDWR FD_CLOEXEC
        SOCK_STREAM
        SO_REUSEADDR,SO_KEEPALIVE,SO_SNDBUF(49152),SO_RCVBUF(49152),↘
            IP_NEXTHOP(0.192.0.0)
        sockname: AF_INET 127.0.0.1  port: 25
   7: S_IFREG mode:0600 dev:288,2 ino:4085770753 uid:0 gid:25 size:33
      O_WRONLY|O_CREAT|O_EXCL|O_LARGEFILE
      /var/run/sendmail.pid
```

Unter Linux erstmal nicht, denn der Kernel exportiert diese Informationen nicht - auch nicht nach `/proc/<pid>/fdinfo`.

Leider!

Aber: SystemTap

SystemTap ist C-artige Skriptsprache zu Instrumentierung des laufenden Kernels.
Aus dem Skript wird ein Kernelmodul erzeugt. Dies wird automatisch geladen und wieder entfernt.

- ist durch den Zugriff auf alle Kernelinterna sehr mächtig
- hohes Risiko durch ein individuelles Kernelmodul
- Red Hat lastig

SystemTap: Möglichkeiten

- `pfiles` Ersatz für Linux
- Verfolgen von Signalen über Prozeßgrenzen
- Auslesen von Kernel-internen Zählern und Statistiken
- ...

Testsetup

`tcpserver.py` und `tcpclient.py` und anschließend die Verbindung testen

```
# ./tcpserver.py
TCPServer Waiting for client on port 5000
I got a connection from  ('127.0.0.1', 50194)
SEND( TYPE q or Q to Quit):Hello


# ./tcpclient.py
RECIEVED: Hello
SEND( TYPE q or Q to Quit):
```

Socket-Optionen anzeigen

```
# ./pfiles.stp 2003
  2003: tcpserver.py
  Current rlimit: 256 file descriptors
   0: S_IFCHR mode:0620 dev:0,10 ino:5 uid:500 gid:500 rdev:136,2
      O_RDWR|O_LARGEFILE
      /dev/pts/2
   1: S_IFCHR mode:0620 dev:0,10 ino:5 uid:500 gid:500 rdev:136,2
      O_RDWR|O_LARGEFILE
      /dev/pts/2
   2: S_IFCHR mode:0620 dev:0,10 ino:5 uid:500 gid:500 rdev:136,2
      O_RDWR|O_LARGEFILE
      /dev/pts/2
   3: S_IFSOCK mode:0777 dev:0,6 ino:19540 uid:500 gid:500 rdev:0,0
      O_RDWR
      socket:[19540]
      SO_KEEPALIVE,SO_TYPE(1),SO_SNDBUF(16384),SO_RCVBUF(87380)
        sockname: AF_INET 0.0.0.0   port: 5000
   4: S_IFSOCK mode:0777 dev:0,6 ino:19542 uid:500 gid:500 rdev:0,0
      O_RDWR
      socket:[19542]
      SO_KEEPALIVE,SO_TYPE(1),SO_SNDBUF(50724),SO_RCVBUF(87552)
        sockname: AF_INET 127.0.0.1   port: 5000
        peername: AF_INET 127.0.0.1   port: 50194
```

strace

```
# strace -e setsockopt ./tcpserver.py
setsockopt(3, SOL_SOCKET, SO_KEEPALIVE, [1], 4) = 0
TCPServer Waiting for client on port 5000
```

- Manpage lsof
- Manpage strace
- Homepage SystemTap

  http://sourceware.org/systemtap/
- SystemTap pfiles

  http://sourceware.org/systemtap/wiki/WSPfiles
- `tcpserver.py`

  http://www.pythonprasanna.com/Papers%20and%20Articles/Sockets/tcpserver_py.txt
- `tcpclient.py`

  http://www.pythonprasanna.com/Papers%20and%20Articles/Sockets/tcpclient_py.txt

Fragen, Anregungen, Meinungen?

Vielen Dank für die Aufmerksamkeit!