Debugging für Sysadmins

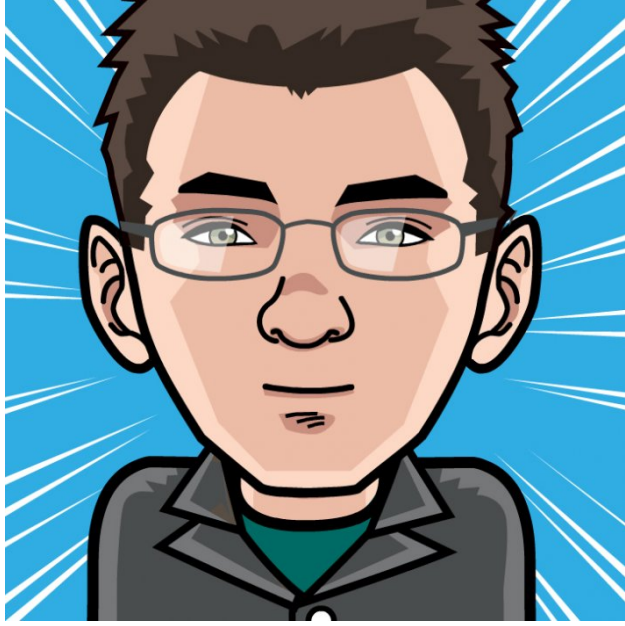
Michael Prokop,
15.04.2023 @ #glt23

SynPro SOLUTIONS

Grazer LINUXTAGE

www.linuxtage.at

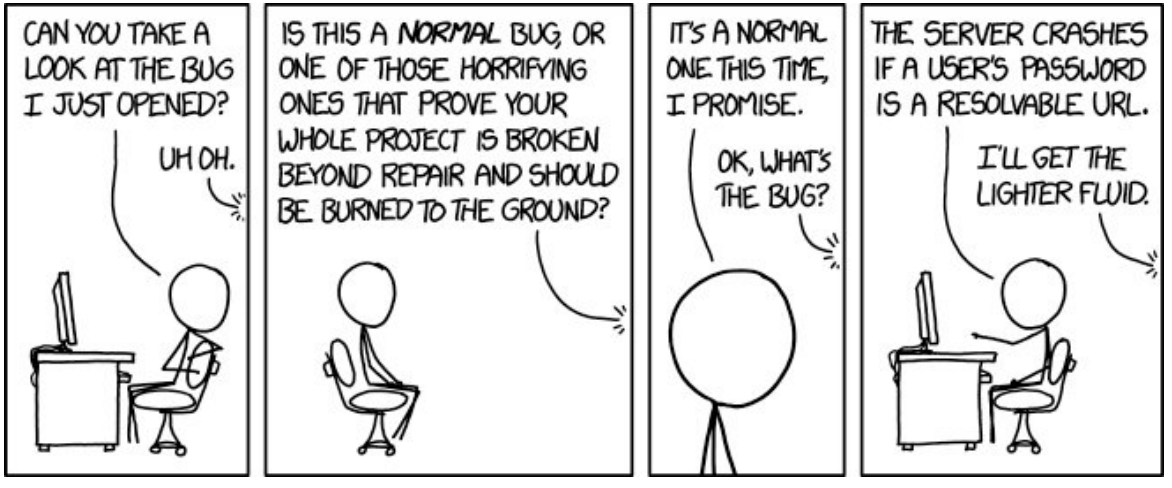
% whoami
mika



- Grml.org Erfinder + Projektleiter
- Debian Developer
- Grml-Forensic (forensische IT-Analysen)
- Geschäftsführer von SynPro Solutions GmbH
  - Strategische IT-Beratung
  - Vermittlung von Best Practices (Trainings/Workshops)
  - Emergency Response
- michael.prokop (at) synpro.solutions

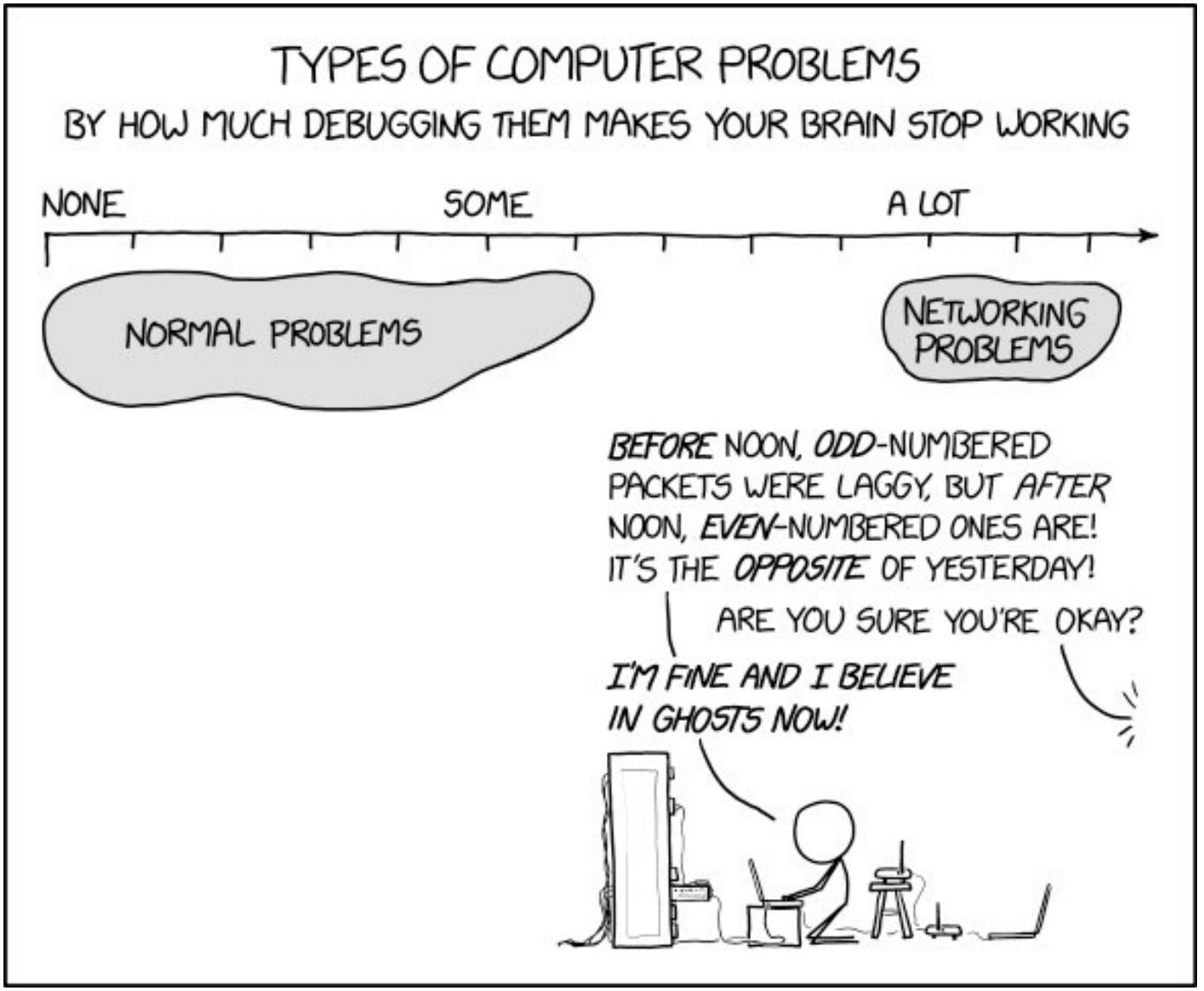
Syadmin's Life – SNAFU

- „The case of the 500-mile email": Ibiblio.org
- „My wife has complained that open office will never print on Tuesdays": Launchpad Bug #255161
- „How we spent two weeks hunting an NFS bug in the Linux kernel“: Gitlab Blog
- RAID-Controller frisst XFS-Daten: Blog-Artikel + Heise iX 10/2021



Quelle: https://xkcd.com/1700/

„LOOK, THE LATENCY FALLS EVERY TIME YOU CLAP YOUR HANDS AND SAY YOU BELIEVE“



Quelle: https://xkcd.com/2259/

Problemerfassung

- Was ist das Problem?
- XY-Problem? („*asking about your attempted (Y) solution rather than your actual problem (X)*“ → ursächliches Problem wird verschleiert)
- Seit wann tritt das Problem auf?
- Ist das denn überhaupt wirklich ein Problem?
- Wie schnell muss das Problem behoben werden?

Problemeingrenzung 1

- Wer wurde auf das Problem aufmerksam?
- Fremdverschulden möglich? Stromausfall, Provider, (D)DoS, Malware,…
- Was ist das gewünschte Verhalten?
  - PEBCAK? (*Problem Exists Between Chair And Keyboard*)
- Was wurde zuletzt gemacht?

Problemeingrenzung 1

- Wer wurde auf das Problem aufmerksam?
- Fremdverschulden möglich? Stromausfall, Provider, (D)DoS, Malware,…
- Was ist das gewünschte Verhalten?
  - PEBCAK? (*Problem Exists Between Chair And Keyboard*)
- Was wurde zuletzt gemacht? **Nichts!**

Problemeingrenzung 1

- Wer wurde auf das Problem aufmerksam?
- Fremdverschulden möglich? Stromausfall, Provider, (D)DoS, Malware,…
- Was ist das gewünschte Verhalten?
  - PEBCAK? (*Problem Exists Between Chair And Keyboard*)
- Was wurde zuletzt gemacht? **Nichts!**
- Was wurde gemacht bevor ***nichts*** gemacht wurde?

Problemeingrenzung 2

- Wenn das Problem dann gelöst sein sollte, woran lässt sich das festmachen? (Monitoring, Kunde,…?)
- Können wir den aktuellen Zustand für später festhalten?
- Kennen wir das Problem bereits? (Doku?)
- Taxonomie → welche Art/Kategorie und welchen Namen hat das Problem? (Hardware, Netzwerk, Software, Sicherheit, Performance,…?)

Patterns

- Ockhams Rasiermesser: Prinzip der einfachsten Theorie
- Hanlons Rasiermesser: geh nicht von Böswilligkeit aus, wenn Inkompetenz genügt
- Confirmation Bias / Bestätigungsfehler: Informationen werden so ausgewählt, dass die eigenen Erwartungen erfüllt werden
- Hindsight Bias / Rückschaufehler: rückblickende Überschätzung der Vorhersagbarkeit → „nachher sind wir immer gescheiter“

Planung / Koordinierung

- Wer *muss* mit ins Boot geholt werden?
- Wer *könnte* mit ins Boot geholt werden?
- Wer muss benachrichtigt werden?
- Wer kommuniziert nach außen hin?
- Wer koordiniert die Leute?
- Externe Hilfe / Fallback-Pläne?
- Kommunikationskanäle + Fallbacks?

Problemanalyse

- Lässt sich das Problem reproduzieren?
  - Gibt es ein Backup mit *getestetem* Restore?
  - Gibt es einen dokumentierten/getesteten Fallbackplan?
  - Passiert das auch in der Referenzumgebung?
  - Gibt’s eine Testumgebung?
- *Wer* kann das Problem reproduzieren?

Quietscheentchen-Debugging

Quelle: https://flic.kr/p/nm2LCE (Tim Reckmann)

Problemanalyse – Fakten, Fakten, Fakten!

- Fakten sammeln: Programmversionen, Runtime (Kernel, Memory,…), Logs, Monitoring, Metriken,…
- Fehlermeldungen/Screenshots/…?
- Englische Fehlermeldung! (LC_ALL=C LANG=C ,…)
  Kein Weltraum links vom Gerät
- Fehlermeldung lesen, und *nochmal* lesen
- Recherche / Suchmaschinen

Problemanalyse – ab ins Detail…

- Falsche Annahmen widerlegen
- Variablen kontrolliert ändern:
  - eines nach dem anderen auschließen
  - Binary Search / Bisect
- Use the source, Luke
  - Debian: debcheckout $PACKAGE || dget $DSC
  - github.com/search + codesearch.debian.net
- Nerd sniping → Abstand gewinnen + Denkpausen einplanen (Spazieren, Duschen, drüber schlafen,…)

Problemanalyse - Doku

- Passt die Doku zum Tool / Tool-Version?
- Schon *während* des Einsatzes dokumentieren:
  - was ist im Nachhinein eventuell noch zu tun
  - für ähnliches Problem beim nächsten Mal
  - für andere Personen (intern / öffentlich)

Vor dem Einsatz spezieller Tools

- It's *always* DNS!
- Uhrzeit/Zeitzone/NTP?
- Netzwerk-Verbindungen, Firewall, MTU?
- AEG probiert?
  - Wichtig: IST-Zustand *vorher* dokumentieren

Vor dem Einsatz spezieller Tools

- Credentials *wirklich* in Ordnung?
- Typos ausgeschlossen? (O vs 0, l vs 1, falsches Tastaturlayout, falsches Encoding,…)
- Berechtigungen (systemd hardening, SELinux, AppArmor, ACLs,…)?
- Environment (`cat /proc/$PID/environ`)?
- Abgelaufene oder ungültige SSL-Zertifikate?
- Laufende Konfiguration != sichtbare Konfiguration? Service-Restart vergessen

Sysadmin’s best friends

- Kenne deine Shell
- Kenne deinen Editor
- Bash, Python, Perl, Ruby,…
- Kenne die klassischen Unix- und Linux-Tools (grep/awk/sed/sort/uniq/head/tail/find/xargs/…)
- SSH (ControlMaster, Port Forwarding, ProxyCommand, Socks Proxy,…)
- Angenehme Arbeitsumgebung

Know your tools

- Generell: wie bekomm ich $TOOL in Verbose- oder Debugging-Modus?
- Lerne neue Tools, aber auch neue Features (ip, lscpu, lsmem, findmnt,… können json-Ausgabe!)
- gute Sammlung an brauchbaren Kommandozeilen und speziell Kommandozeilen-Optionen
- Sicherstellen, dass die klassischen Tools auf allen Systemen zur Verfügung stehen (Konfigurationsmanagement!)
- Netzwerk-Tools mit `-n` aufrufen

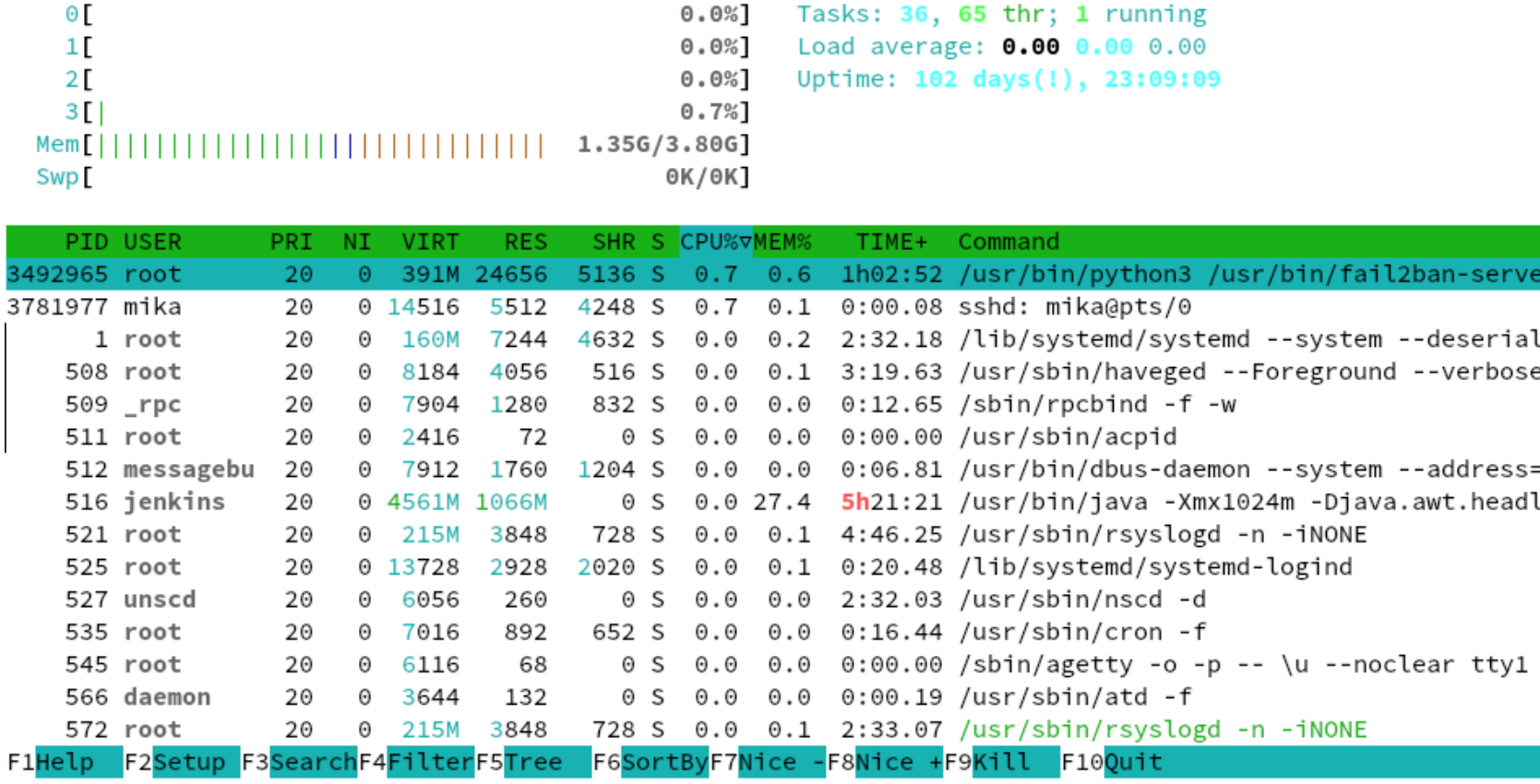
Schneller Überblick: System

```
uptime && uptime -s
df -h [-l]
free -m
top || htop
ps auxwwf
dmesg -T | tail
systemctl --failed
journalctl -xe [--priority=3]
iostat -x 5 [-z]
```

Schneller Überblick: Netzwerk

```
ip link
ip a
ip r
ip route get $IP
ss -tulpnO [-e]
ip rule list
ping -D [-s $PKTSIZE] [-c $COUNT] [-4|-6] $HOST
dig $HOST [$TYPE] [+short]
```

htop



- e: show process environment
- H: hide/show user process threads
- K: hide/show kernel threads
- l: list open files with lsof
- s: trace syscalls with strace

systemd

```
systemctl --failed
systemctl cat 'foo*'
sudo systemctl status $PID
sudo journalctl -u "foo*" --since 2023-04-15 -f
coredumpctl debug
```

strace – wichtige™ Optionen

```
-f              → follow forks
-o $filename    → write trace output to file
-p $pid         → attach to process ID
-s $strsize     → maximum string size to print
-e $expr        → which events to trace
-Z              → print only syscalls that
                  returned with an error code
```

strace - Beispiele

```
strace -f -o strace.log …
strace -f -eopen,close,read,write …
strace -f -e trace=%process …
strace -f -e %net …
strace -f -Z -e %file …
strace -f -s500 -p $PID
strace -f … 2>&1 | grep FOO
```

Real-World Problem

- Problembeschreibung: ein Python-Skript mit cx_Oracle-Bibliothek funktioniert auf einem Debian-System nicht
- Fehlermeldung: „*cx_Oracle.DatabaseError: DPI-1047: Cannot locate a 64-bit Oracle Client library: "libclntsh.so: cannot open shared object file: No such file or directory"*“

Real-World Debugging - Reproduce

```
synpromika@foo ~ % cat demo.py
import cx_Oracle
connection = cx_Oracle.connect(user="foo",
password="bar", dsn="localhost/orclpdb1")

synpromika@foo ~ % export DPI_DEBUG_LEVEL=64
synpromika@foo ~ % python3 ./demo.py
[…]
  cx_Oracle.DatabaseError: DPI-1047: Cannot locate a 64-
bit Oracle Client library: "libclntsh.so: cannot open
shared object file: No such file or directory". […]
```

Real-World Debugging – strace FTW!

```
synpromika@foo ~ % strace -f -Z python3 ./demo.py
[…]
openat(AT_FDCWD, "/opt/instantclient_21_3/libclntsh.so.21.1", O_RDONLY|
O_CLOEXEC) = -1 EACCES (Permission denied)
[…]

synpromika@foo ~ % ls -la /opt/instantclient_21_3/libclntsh.so.21.1
-rw-r----- 1 root root 83460712 Jul 27 14:11
/opt/instantclient_21_3/libclntsh.so.21.1

synpromika@foo ~ % sudo chmod 644 /opt/instantclient_21_3/libclntsh.so.21.1
```

Netzwerk-Tools - Mitlauschen

```
tcpdump -n -i $DEV -s 65535 not port 22

tcpdump|tshark […] -w capture.pcap
wireshark capture.pcap || tshark -r capture.pcap

tshark […] -f “host $IP“
tshark […] -f ‘port not 53 and not arp‘
tshark […] -r capture.pcap -Y ‘tcp.port == 80‘
tshark […] -r capture.pcap -Y ‘ip.ttl > 10‘

tshark -n -d tcp.port==8080,http --color [--json]
tshark -n -d udp.port==1053,dns  --color [--json]
```

Netzwerk-Tools - tshark

```
% tshark -n -Y 'dns.qry.name ~ linuxtage.at'
[…]
 1721 29.437507385 192.168.88.42 → 192.168.88.1
DNS 72 Standard query 0x2fcc A linuxtage.at
 1722 29.437573481 192.168.88.42 → 192.168.88.1
DNS 72 Standard query 0x5120 HTTPS linuxtage.at
 1724 29.467845232 192.168.88.1 → 192.168.88.42
DNS 88 Standard query response 0x2fcc A
linuxtage.at A 91.151.18.164
[…]
```

Netzwerk-Tools - ss/iproute2

```
% sudo ss -tulpn0 -o state all '( sport = :22 )'
```

```
mika@lunge ~ % sudo ss -tulpn -o state all '( sport = :22 )'
Netid State  Recv-Q Send-Q Local Address:Port Peer Address:PortProcess
tcp   LISTEN 0      128          0.0.0.0:22        0.0.0.0:*    users:(("sshd",pid=4102430,fd=3))
tcp   LISTEN 0      128             [::]:22           [::]:*    users:(("sshd",pid=4102430,fd=4))
mika@lunge ~ % sudo ss -tulpn -o state '( sport = :22 )'
completing state
all             CLOSED          connected       FIN-WAIT-2      synchronized    TIME-WAIT
big             CLOSE-WAIT      ESTABLISHED     LAST-ACK        SYN-RECV
bucket          CLOSING         FIN-WAIT-1      LISTENING       SYN-SENT
```

Netzwerk-Tools - mtr

```
% mtr --aslookup --report --report-cycles=10 linuxtage.at
Start: Thu Apr 13 13:42:11 2023
HOST: demo                          Loss%   Snt   Last   Avg  Best  Wrst StDev
  1. AS42473 202.61.209.98          0.0%    10    4.0   2.7   0.5   8.2   2.7
  2. AS47147 ae1-0.bbr01.anx88.vi   0.0%    10    0.8   5.7   0.6  49.4  15.4
  3. AS47147 nessus-gw.bbr01.anx8   0.0%    10    1.2   1.5   0.8   2.7   0.6
  4. AS47692 91.151.18.164          0.0%    10    0.9   1.0   0.9   1.1   0.0
```

Netzwerk-Tools – Misc

```
curl [-6] --verbose --connect-to \
example.org:443:$IP:443 https://example.org

traceroute -n -T -p 443 $HOST

hping3 -c 1 --udp -d 2000 $HOST

echo QUIT | openssl s_client -connect \
example.org:443 2>/dev/null | \
openssl x509 -noout -dates

echo QUIT | openssl s_client -starttls smtp \
-connect mail.example.org:smtp 2>/dev/null | \
openssl x509 -noout -dates
```

Misc Tricks - grep

Return/exit on first match (speedup!):

```
grep -m1 …
```

Misc Tricks - diff/git

word-diff - braucht kein .git!

```
git diff --word-diff $f1 $f2
```

```
mika@lunge /tmp/tmp.hCbxoYYpEP % git diff --word-diff interfaces.broken interfaces
diff --git interfaces.broken interfaces
index b4fc2a5..224fe43 100644
--- interfaces.broken
+++ interfaces
@@ -1,3 +1,3 @@
auto vmbr1
iface vmbr1 inet static
        address [-10.42.15.2/24-]{+10.42.15.3/24+}
```

Misc Tricks - du/sort

Warum ist das Dateisystem voll?

```
du -xha --max-depth=1 / | sort -h
```

Vieeeele weitere Tools…

- BPF-Tools (bpftrace / bpfcc-tools)
- linux-perf
- ltrace
- swaks
- socat
- gdb + debuginfod
- pt-mysql-summary, mysqlreport, apachetop,…
- …

Nach dem Problem ist *vor* dem Problem

- System wieder sauber zurückgelassen?
  - Loglevel wieder zurückgestellt?
  - Debug-/Tmpfiles weggeräumt?
  - Temporäre Firewall-Regeln zurückgesetzt?
  - Sind System-Änderungen im Cfgmgt?
  - …
- Dokumentation (review, ergänzen,…)
- Post Mortem?
- Über das Problem reden, schreiben, bloggen

Nach dem Problem ist *vor* dem Problem

- Was lässt sich aus dem Vorfall lernen?
- Wie lässt sich so ein Problem in Zukunft vermeiden?
- Sind Monitoring, Logging,… adäquat?
- Konstruktive Fehlerkultur / Lernkultur!

Wie werden wir besser?

- Test- und Trainingsumgebung
- Support machen
  - speziell: selbst kein direkter Zugriff auf das betroffene System
- Öffentliche Post-Mortems lesen
- Wer öfter schwierige Probleme löst, kommt auch öfter bei interessanten Problemen zum Einsatz

^D
Connection closed.

https://michael-prokop.at/blog/
michael.prokop (at) synpro.solutions

SynPro SOLUTIONS

Grazer LINUXTAGE

www.linuxtage.at