



Troubleshooting Tools

for Surrey LUG

© Adrian Bridgett <adrian@smop.co.uk>

Presentation released under GPL v2

Introduction



- Everyone can troubleshoot :-)
- Like most things you need:
 - knowledge
 - experience
 - luck
- I hope to impart some of the first two
 - there's a reason I'm not a betting man

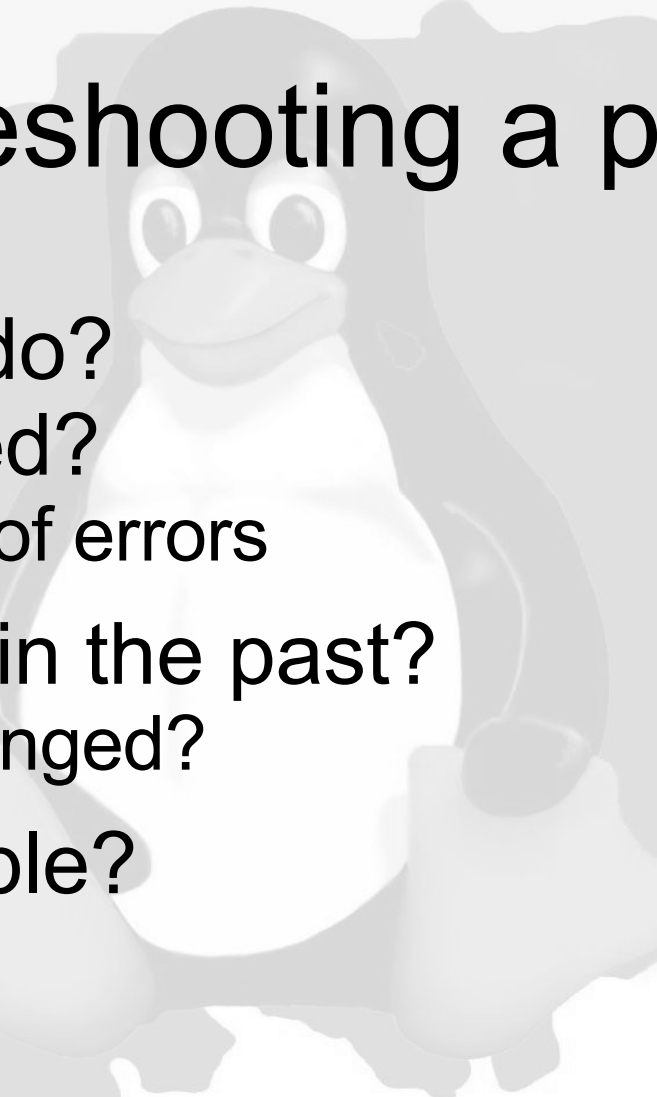
Contents



- Troubleshooting a program
 - checking the web
 - debugging
 - getting help

Troubleshooting a program

- what did you do?
- what happened?
 - make a note of errors
- has it worked in the past?
 - what has changed?
- is it reproducible?



Checking the Web



- Use Google to see if it is known/resolved
- Is there a later version available?
 - might not be packaged yet
 - check changelog/NEWS file
- Check bug systems
 - Debian: <http://bugs.debian.org> or reportbug
 - Ubuntu: <https://launchpad.net/distros/ubuntu/+bugs>
 - Fedora: <http://www.fedora.info/bugzilla/>

Debugging - application

- Check manpage for “verbose” or “debug” options
 - normally -v or -d
 - can often be specified several times
 - e.g. -vvvv for really verbose messages
- Redirect output to capture it:
 - program name > logfile 2>&1
 - 2>&1 makes “standard error” message goto the same place that we just redirected “standard output” to

Debugging – strace example

- traces system calls and signals
- `strace -f -e trace=file -o logfile cat /etc/shadow`
 - follow forks, just filesystem commands
 - very useful to check for permission denied

```
15951 execve("/bin/cat", ["cat", "/etc/shadow"], [/* 63 vars */]) = 0
15951 access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
15951 access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
15951 open("/etc/ld.so.cache", O_RDONLY) = 3
15951 access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
15951 open("/lib/tls/i686/cmov/libc.so.6", O_RDONLY) = 3
15951 open("/usr/lib/locale/locale-archive", O_RDONLY|O_LARGEFILE) = 3
15951 open("/etc/shadow", O_RDONLY|O_LARGEFILE) = -1 EACCES (Permission denied)
...
```

Debugging – strace advanced

- Can also log time spent in call:
 - `strace -T`
- Or just timestamp calls:
 - `strace -tt`
- Or attach to running programs:
 - `strace -p program name`
- Strings only have first 32 characters printed
 - filenames don't count (always printed in full)
 - `-s 500` will increase to 500 characters

Debugging – strace warnings

- Warnings:
 - logs get real big, real fast
 - strace can seriously slow a program
 - detaching from a program (killing strace) can cause problems – particularly with multithreaded apps

Debugging – shell

- Use “-x” option to print lines before they are executed
 - “set +x” in scripts
 - “bash -x program” is easier

Debugging - print



- Anyone can program
 - well, a little anyway
- Shell: `echo "hello $filename"`
- Perl: `print "hello $filename"`
 - `print "hello @filenames"` for an array...
- Python: `print "hello", filename`
- Ruby: `print "hello #{filename}"`
 - or possibly `print "hello #{@filename}"`

Debugging – netstat

- Cannot talk to something on your machine?
- Use “netstat” to test:

- netstat -pant (easy to remember)

```
tcp          0      0 0.0.0.0:25      0.0.0.0:*      LISTEN      2312/inetd
tcp          0      0 0.0.0.0:80      0.0.0.0:*      LISTEN      311/apache2
```

- netstat -panu for UDP ports (e.g. DNS, NTP)
 - “-p” will print the program name if possible

Debugging - tcpdump

- Logs network traffic
- Use “netstat -nr” or “ifconfig” to find the device
 - eth0, eth1, wlan0 most common
 - lo for most local traffic
- tcpdump -i eth0 -s 1500 -w filename [options]
 - must run as root
 - increase size (-s) if you need the full packets
 - 1500 for most networks
 - 16436 for lo device

Debugging – tcpdump II

- common options:
 - host *machine*
 - port *portname*
 - port *portnumber*
 - e.g. “not port 22” to exclude ssh traffic
- use wireshark to view captured files
 - wireshark was formerly known as ethereal
 - wireshark can also capture packets
 - I'm just used to tcpdump

Getting help



- Mailing lists or forums
 - either for the project
 - or your local LUG
 - considered polite to at least Google a bit first
- IRC – Internet Relay Chat
- File a bug
 - include as much info as you can
 - versions, error messages, what you did
 - debugging output appreciated
 - if only because they know you put some effort in

Summary



- These are the tools I use for 90% of problems
- “there was an error message but I didn't write it down” isn't helpful!
 - programmers generally the worst offenders
- Debugging remotely is hard – particularly by phone
- The more information you can give, the faster the problem will be fixed
 - particularly if you can Google and find the answer!

A cartoon penguin with a white body, black head, and large eyes is standing on a map of the British Isles. The map is rendered in a light gray tone, showing the outlines of Great Britain and Ireland. The penguin is positioned in the center of the map, facing forward.

Thank you!

Adrian Bridgett
<adrian@smop.co.uk>