
ORACLE®

ORACLE

Stupid Wireshark Tricks

Chuck Lever
Consulting Member of Technical Staff

General Comments

- Stability
  - Wireshark crashes
  - Dissectors use heuristics, get confused
- Missing features for us
  - Parts of NFSv4.1 and pNFS
  - FedFS ADMIN
  - Implementation-specific decoders for client IDs, state IDs, file handles, etc.
- When in doubt, try wireshark before tshark
  - Enormous flexibility and rich feature set
  - The GUI can guide you through esoteric features

ORACLE

Filter Types

Capture Filters

- Same as `tcpdump` filters
- `man pcap-filter(7)` on Linux
- Set before capturing
- Course-grained, but useful for reducing volume of captured data
- Examples
  - `tcp dst port <nnnn>`
  - `host <aa.bb.cc.dd>`

Display Filters

- Specific to `wireshark`
- `man wireshark-filter(4)` on Linux
- Can be changed while capturing, or while displaying previously captured frames
- Finer-grained than capture filters
- Examples
  - `tcp.port == <nnnn>`
  - `ip.addr == <aa.bb.cc.dd>`
  - `(nfs) && (rpc.programversion == 4)`

Snoop Idioms

Snoop Idioms

- Capture any traffic involving <host>

```
$ snoop <host>
```

is equivalent to:

```
$ tshark host <host>
```

Snoop Idioms

- Capture traffic between <host1> and <host2>

```
$ snoop <host1> <host2>
```

is equivalent to:

```
$ tshark “host <host1> && host <host2>”
```

Snoop Idioms

- Capture all packets from port <nn> on host <host>

```
$ snoop <host> from port <nn>
```

is equivalent to:

```
$ tshark “host <host> && port <nn>”
```

Snoop Idioms

- Display packets containing NFS payload

```
$ snoop rpc nfs
```

is equivalent to:

```
$ tshark -R nfs
```

Snoop Idioms

- Read from <capture_file> rather than reading live traffic

```
$ snoop -i <capture_file>
```

is equivalent to:

```
$ tshark -r <capture_file>
```

Snoop Idioms

- Write raw packets to <capture_file>

```
$ snoop -o <capture_file>
```

is equivalent to:

```
$ tshark -w <capture_file>
```

Snoop Idioms

- Display frames <fr1> through <fr2>

```
$ snoop -i <capture_file> \
    -p <fr1>,<fr2>
```

is equivalent to:

```
$ tshark -r <capture_file> \
    -R "(frame.number >= <fr1>) && \
        (frame.number <= <fr2>)"
```

Snoop idioms

- Display details of only frame <fr>

```
$ snoop -i <capture_file> -p <fr> -v
```

is equivalent to:

```
$ tshark -r <capture_file> -V \
    -R "frame.number == <fr>"
```

Snoop idioms

- Read from network interface <intf> instead of the default

```
$ snoop -d <intf>
```

  is equivalent to:

```
$ tshark -i <intf>
```

- Capture traffic on all interfaces:

```
$ tshark -i any
```

Snoop idioms

- Capture only the first 200 bytes of each frame

```
$ snoop -s 200 -o <capture_file>
```

is equivalent to:

```
$ tshark -s 200 -w <capture_file>
```

Snoop Idioms

- Display frames with absolute time

```
$ snoop -t a
```

is equivalent to:

```
$ tshark -t a
```

Snoop Idioms

- Disable reverse-mapping IP and MAC addresses

```
$ snoop -r
```

is equivalent to:

```
$ tshark -n
```

Extended Features

Reducing Capture Volume

- Capture autostop
  - -a duration:<seconds>
  - -a filesize:<kilobytes>
  - -a files:<count>
- Capture ring buffer
  - -b duration:<seconds>
  - -b filesize:<kilobytes>
  - -b files:<count>
- Capture packet count
  - -c <count>

Reducing Capture Volume

- Disable promiscuous mode
  - -p
- Decrease snaplen
  - -s 200
  - *NB:* IPoIB can generate frames larger than 65535

Fine-tuning tshark's Output

- -V - very verbose
- -d - decode as
  - Example: "`-d tcp.port==xxxx,http`"
- -z - statistics
  - Example: "`-q -z rpc,programs`"

Fine-tuning tshark's output

- -T - set output format
  - pdml - detailed output in XML format
  - psml - summary output in XML format
  - ps - PostScript
  - text - default text on standard output
  - fields - comma-separated fields (used with -e)
- -e field1 -e field2 -e ....
  - Examples: frame.number, ip.addr, nfs.read.data_length
  - "-G fields" generates glossary of field names

ORACLE

For More Information

http://www.wireshark.org/docs/

Wireshark Demonstration

ORACLE IS THE INFORMATION COMPANY