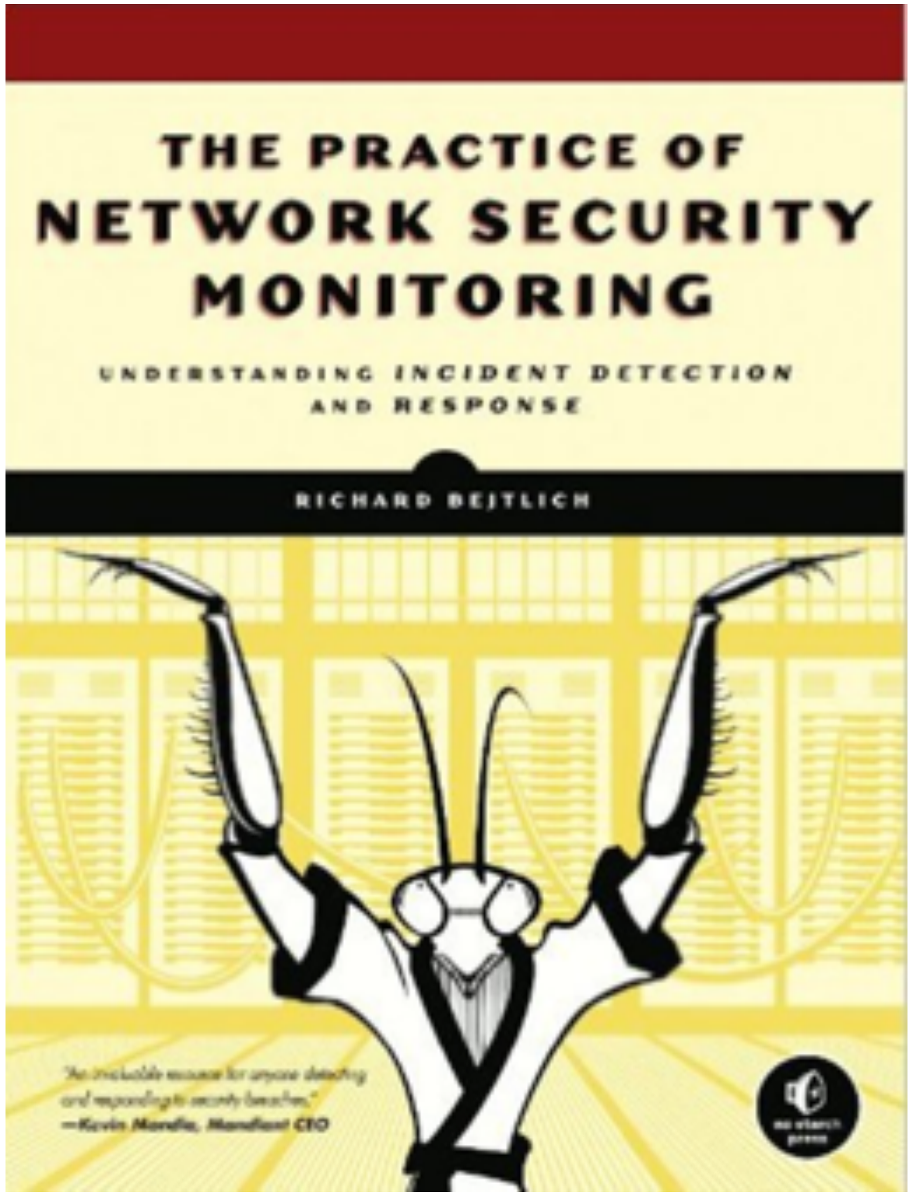
CNIT 50: Network Security Monitoring

6 Command Line Packet Analysis Tools


THE PRACTICE OF NETWORK SECURITY MONITORING
UNDERSTANDING INCIDENT DETECTION AND RESPONSE
RICHARD BEJTLICH
"An invaluable resource for anyone detecting and responding to security breaches."
—Kevin Mandia, Mandiant CEO
no starch press

Topics

- **SO Tool Categories**
- **Running Tcpdump**
- **Using Dumpcap and Tshark**
- **Running Argus and the Ra Client**

SO Tool Categories

Three Types of Tools

- Data presentation
- Data collection
- Data delivery

Data Presentation Tools

- Packet Analysis Tools
  - Read traffic from a live interface or from a saved PCAP file
  - Command-line: **tcpdump**, **Tshark** (with **Dumpcap**), and **Argus Ra Client**
  - Graphical interface: **Wireshark**, **Xplico**, and **NetworkMiner** (see Ch 7)

NSM Consoles

- Gateways to NSM data
- **Squil**, **Squert**, and **ELSA** (see Ch 8)
  - Text discusses **Snorby** but it's abandoned and no longer included in Security Onion
    - Links Ch 1e, 1f

Data Collection Tools

- These applications collect and generate the NSM data available to the presentation tools
- **Argus server**, **Netsniff-ng**, **PRADS**, **Snort**, **Suricata**, and **Bro**

Argus and PRADS

- **Argus server** and **PRADS** create and store their own form of session data

- **Argus** uses a proprietary binary format suited for rapid command-line mining

- **PRADS** data is best read through an NSM console

Netsniff-ng

- Simply writes full-content data to disk in pcap format

Snort and Suricata

- Network intrusion detection systems (NIDS)
- Inspect traffic and write alerts
- According to signatures deployed with each tool

Bro

- Observes and interprets traffic that has been generted and logged in a variety of NSM datatypes

Data Delivery Tools

- Middleware between the data presentation and data collection tools

- **PulledPork** manages IDS rules

- **Barnyard2** manages alert processing

- **Capme** manages pcap access

Squil Agents

- Shuttle data from the collection tools to the presentation software
- **pcap_agent** and **snort_agent**
- **Apache** web server
- **MySQL** database
- **Sphinx** index application

Integrating Tools

- Integrate host-centric analysis analysis features
- **OSSEC** host IDS
- **Syslog-ng** for transport and aggregation of log messages

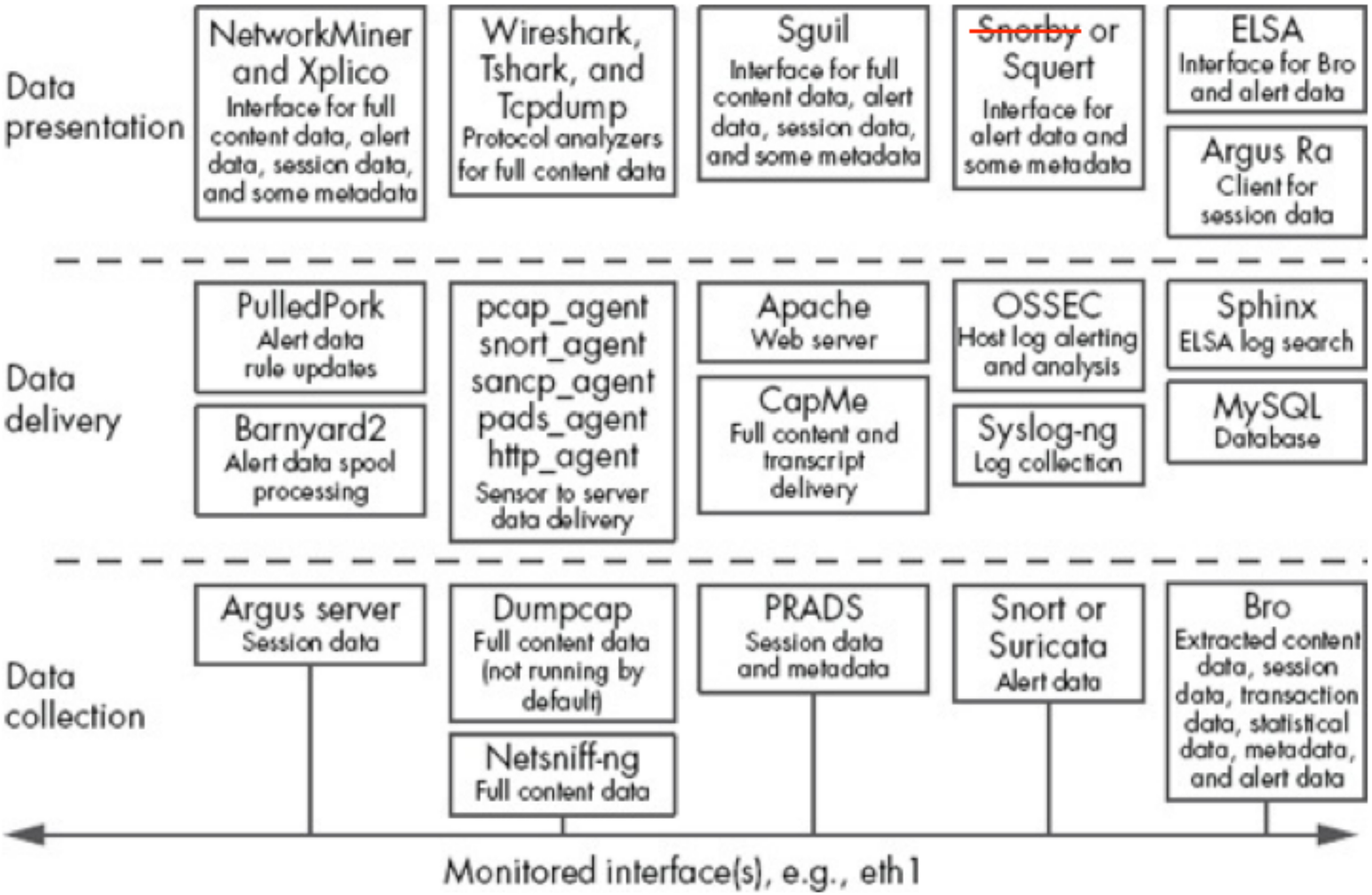

Data presentation
NetworkMiner and Xplico
Interface for full content data, alert data, session data, and some metadata
Wireshark, Tshark, and Tcpdump
Protocol analyzers for full content data
Sguil
Interface for full content data, alert data, session data, and some metadata
~~Snorby~~ or Squert
Interface for alert data and some metadata
ELSA
Interface for Bro and alert data
Argus Ra
Client for session data
Data delivery
PulledPork
Alert data rule updates
Barnyard2
Alert data spool processing
pcap_agent
snort_agent
sancp_agent
pads_agent
http_agent
Sensor to server data delivery
Apache
Web server
CapMe
Full content and transcript delivery
OSSEC
Host log alerting and analysis
Syslog-ng
Log collection
Sphinx
ELSA log search
MySQL
Database
Data collection
Argus server
Session data
Dumpcap
Full content data (not running by default)
Netsniff-ng
Full content data
PRADS
Session data and metadata
Snort or Suricata
Alert data
Bro
Extracted content data, session data, transaction data, statistical data, metadata, and alert data
Monitored interface(s), e.g., eth1

Figure 6-1. Core SO tools

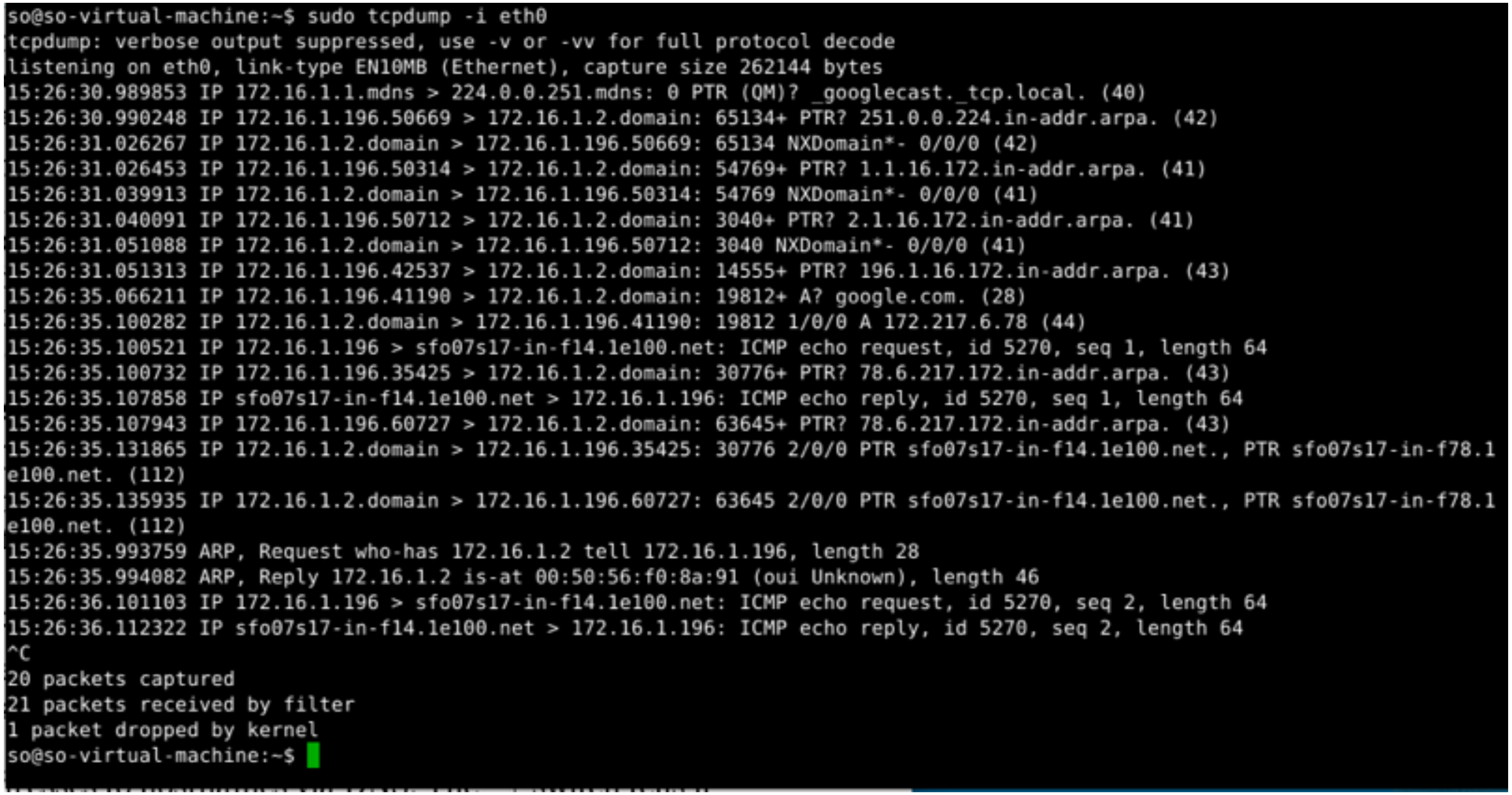
Running Tcpdump

Tcpdump

- Protocol analyzer: understands layers of networking
- Included in SO but not running by default
- Often used to analyze pcaps in **/nsn/sensor_data/<sensorname>/dailylogs**
- Can also collect live data

Basic Usage

- Requires **sudo**
- Specify interface with **-i**

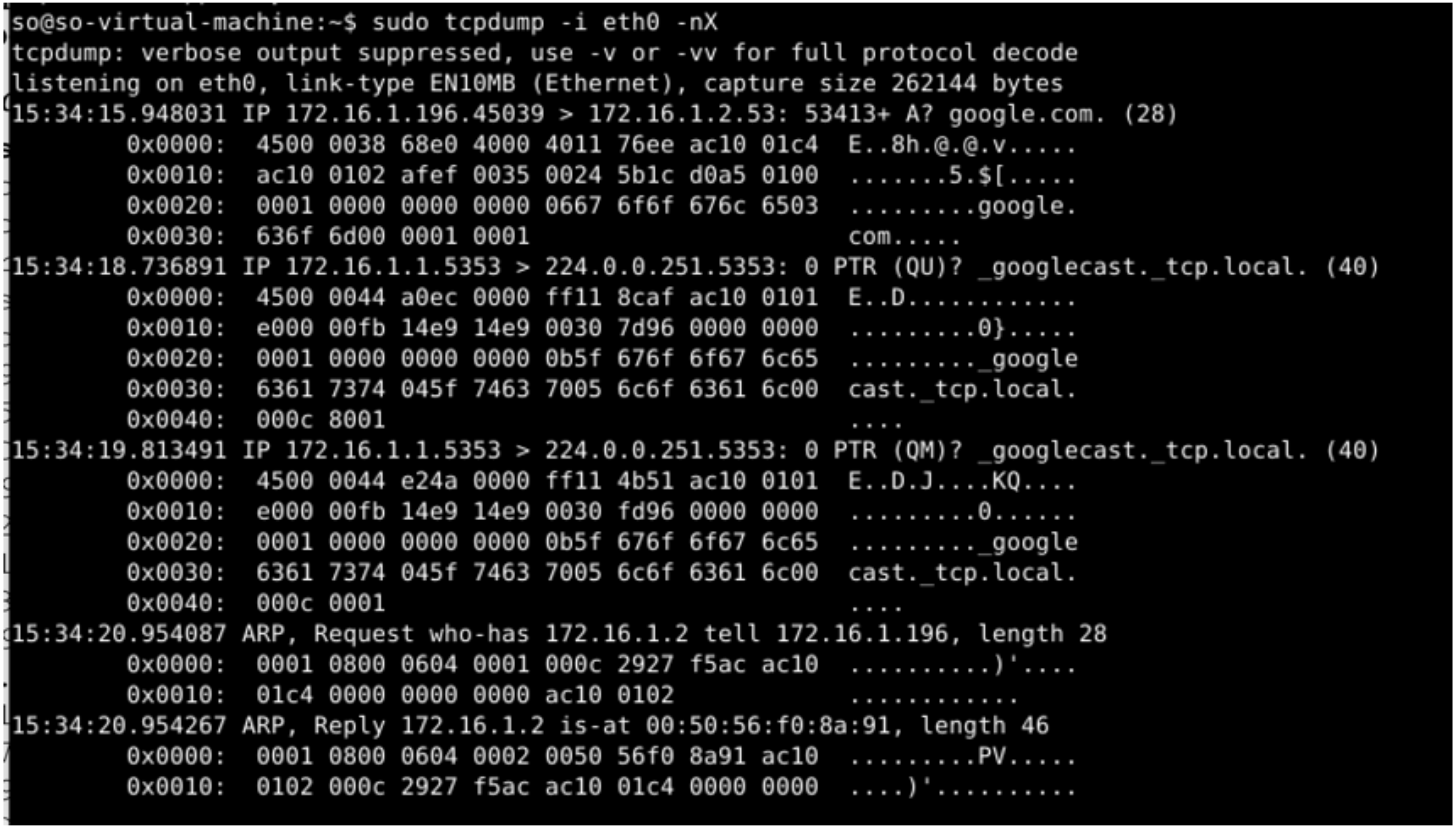


```
so@so-virtual-machine:~$ sudo tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
15:26:30.989853 IP 172.16.1.1.mdns > 224.0.0.251.mdns: 0 PTR (QM)? _googlecast._tcp.local. (40)
15:26:30.990248 IP 172.16.1.196.50669 > 172.16.1.2.domain: 65134+ PTR? 251.0.0.224.in-addr.arpa. (42)
15:26:31.026267 IP 172.16.1.2.domain > 172.16.1.196.50669: 65134 NXDomain*- 0/0/0 (42)
15:26:31.026453 IP 172.16.1.196.50314 > 172.16.1.2.domain: 54769+ PTR? 1.1.16.172.in-addr.arpa. (41)
15:26:31.039913 IP 172.16.1.2.domain > 172.16.1.196.50314: 54769 NXDomain*- 0/0/0 (41)
15:26:31.040091 IP 172.16.1.196.50712 > 172.16.1.2.domain: 3040+ PTR? 2.1.16.172.in-addr.arpa. (41)
15:26:31.051088 IP 172.16.1.2.domain > 172.16.1.196.50712: 3040 NXDomain*- 0/0/0 (41)
15:26:31.051313 IP 172.16.1.196.42537 > 172.16.1.2.domain: 14555+ PTR? 196.1.16.172.in-addr.arpa. (43)
15:26:35.066211 IP 172.16.1.196.41190 > 172.16.1.2.domain: 19812+ A? google.com. (28)
15:26:35.100282 IP 172.16.1.2.domain > 172.16.1.196.41190: 19812 1/0/0 A 172.217.6.78 (44)
15:26:35.100521 IP 172.16.1.196 > sfo07s17-in-f14.1e100.net: ICMP echo request, id 5270, seq 1, length 64
15:26:35.100732 IP 172.16.1.196.35425 > 172.16.1.2.domain: 30776+ PTR? 78.6.217.172.in-addr.arpa. (43)
15:26:35.107858 IP sfo07s17-in-f14.1e100.net > 172.16.1.196: ICMP echo reply, id 5270, seq 1, length 64
15:26:35.107943 IP 172.16.1.196.60727 > 172.16.1.2.domain: 63645+ PTR? 78.6.217.172.in-addr.arpa. (43)
15:26:35.131865 IP 172.16.1.2.domain > 172.16.1.196.35425: 30776 2/0/0 PTR sfo07s17-in-f14.1e100.net., PTR sfo07s17-in-f78.1
e100.net. (112)
15:26:35.135935 IP 172.16.1.2.domain > 172.16.1.196.60727: 63645 2/0/0 PTR sfo07s17-in-f14.1e100.net., PTR sfo07s17-in-f78.1
e100.net. (112)
15:26:35.993759 ARP, Request who-has 172.16.1.2 tell 172.16.1.196, length 28
15:26:35.994082 ARP, Reply 172.16.1.2 is-at 00:50:56:f0:8a:91 (oui Unknown), length 46
15:26:36.101103 IP 172.16.1.196 > sfo07s17-in-f14.1e100.net: ICMP echo request, id 5270, seq 2, length 64
15:26:36.112322 IP sfo07s17-in-f14.1e100.net > 172.16.1.196: ICMP echo reply, id 5270, seq 2, length 64
^C
20 packets captured
21 packets received by filter
1 packet dropped by kernel
so@so-virtual-machine:~$
```

Other Useful Switches

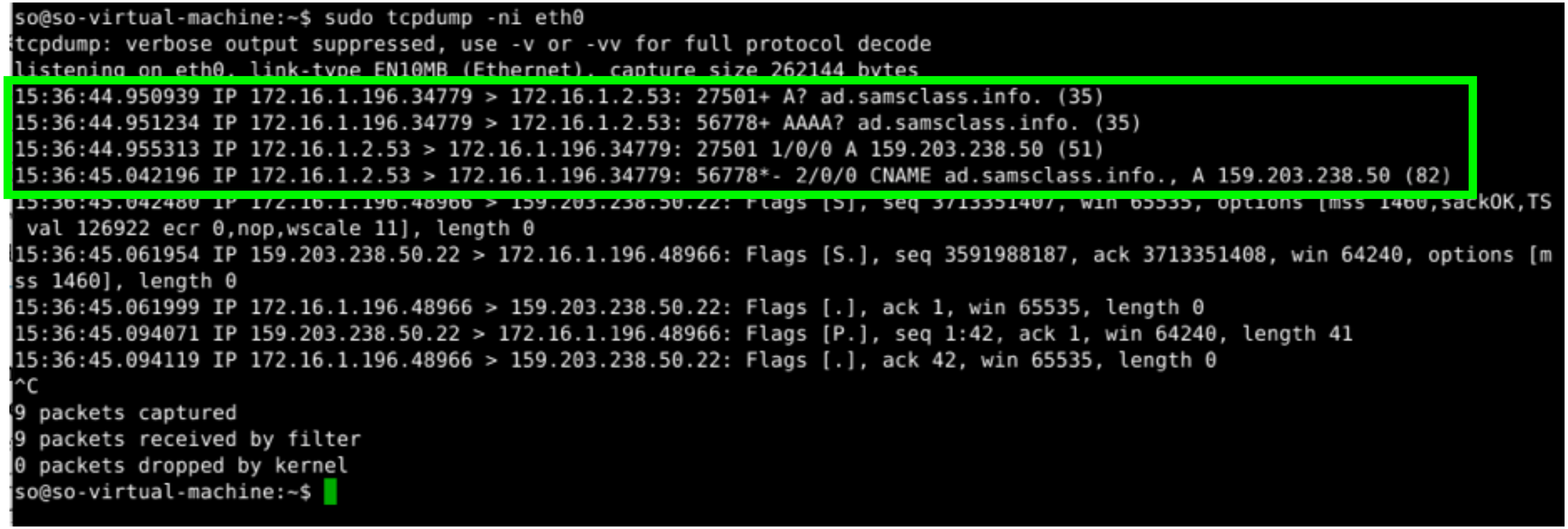
- **-n** Don't resolve names
- **-s *#*** Adjust "snaplength" -- Number of bytes to collect (default is 68 bytes for IPv4)
- **-c *count*** Only collect *count* packets (0 for all data)
- **-X** Print out packet bytes
- **-w *filename.pcap*** Write PCAP file

-X



```
so@so-virtual-machine:~$ sudo tcpdump -i eth0 -nX
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
15:34:15.948031 IP 172.16.1.196.45039 > 172.16.1.2.53: 53413+ A? google.com. (28)
        0x0000:  4500 0038 68e0 4000 4011 76ee ac10 01c4  E..8h.@.@.v.....
        0x0010:  ac10 0102 afef 0035 0024 5b1c d0a5 0100  .......5.$[.....
        0x0020:  0001 0000 0000 0000 0667 6f6f 676c 6503  .........google.
        0x0030:  636f 6d00 0001 0001                      com.....
15:34:18.736891 IP 172.16.1.1.5353 > 224.0.0.251.5353: 0 PTR (QU)? _googlecast._tcp.local. (40)
        0x0000:  4500 0044 a0ec 0000 ff11 8caf ac10 0101  E..D............
        0x0010:  e000 00fb 14e9 14e9 0030 7d96 0000 0000  .........0}.....
        0x0020:  0001 0000 0000 0000 0b5f 676f 6f67 6c65  ........._google
        0x0030:  6361 7374 045f 7463 7005 6c6f 6361 6c00  cast._tcp.local.
        0x0040:  000c 8001                                ....
15:34:19.813491 IP 172.16.1.1.5353 > 224.0.0.251.5353: 0 PTR (QM)? _googlecast._tcp.local. (40)
        0x0000:  4500 0044 e24a 0000 ff11 4b51 ac10 0101  E..D.J....KQ....
        0x0010:  e000 00fb 14e9 14e9 0030 fd96 0000 0000  .........0......
        0x0020:  0001 0000 0000 0000 0b5f 676f 6f67 6c65  ........._google
        0x0030:  6361 7374 045f 7463 7005 6c6f 6361 6c00  cast._tcp.local.
        0x0040:  000c 0001                                ....
15:34:20.954087 ARP, Request who-has 172.16.1.2 tell 172.16.1.196, length 28
        0x0000:  0001 0800 0604 0001 000c 2927 f5ac ac10  ..........)'....
        0x0010:  01c4 0000 0000 0000 ac10 0102            ............
15:34:20.954267 ARP, Reply 172.16.1.2 is-at 00:50:56:f0:8a:91, length 46
        0x0000:  0001 0800 0604 0002 0050 56f0 8a91 ac10  .........PV.....
        0x0010:  0102 000c 2927 f5ac ac10 01c4 0000 0000  ....)'..........
```

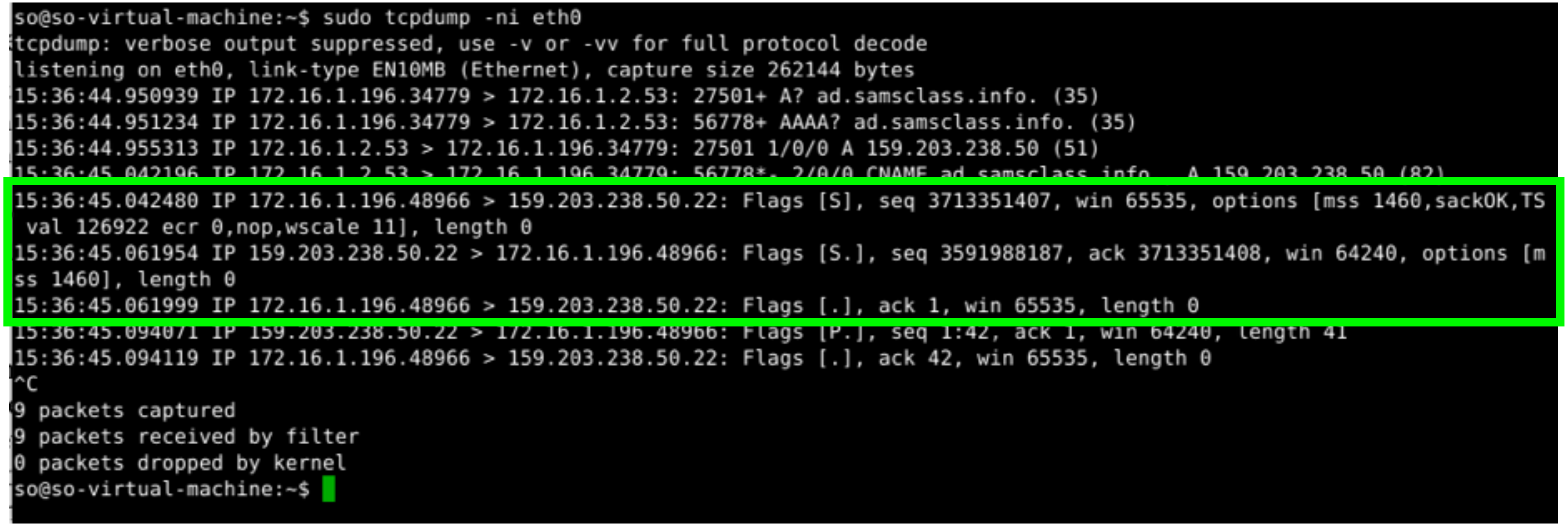
DNS Query & Reply



```
so@so-virtual-machine:~$ sudo tcpdump -ni eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
15:36:44.950939 IP 172.16.1.196.34779 > 172.16.1.2.53: 27501+ A? ad.samsclass.info. (35)
15:36:44.951234 IP 172.16.1.196.34779 > 172.16.1.2.53: 56778+ AAAA? ad.samsclass.info. (35)
15:36:44.955313 IP 172.16.1.2.53 > 172.16.1.196.34779: 27501 1/0/0 A 159.203.238.50 (51)
15:36:45.042196 IP 172.16.1.2.53 > 172.16.1.196.34779: 56778*- 2/0/0 CNAME ad.samsclass.info., A 159.203.238.50 (82)
15:36:45.042480 IP 172.16.1.196.48966 > 159.203.238.50.22: Flags [S], seq 3713351407, win 65535, options [mss 1460,sackOK,TS
 val 126922 ecr 0,nop,wscale 11], length 0
15:36:45.061954 IP 159.203.238.50.22 > 172.16.1.196.48966: Flags [S.], seq 3591988187, ack 3713351408, win 64240, options [m
ss 1460], length 0
15:36:45.061999 IP 172.16.1.196.48966 > 159.203.238.50.22: Flags [.], ack 1, win 65535, length 0
15:36:45.094071 IP 159.203.238.50.22 > 172.16.1.196.48966: Flags [P.], seq 1:42, ack 1, win 64240, length 41
15:36:45.094119 IP 172.16.1.196.48966 > 159.203.238.50.22: Flags [.], ack 42, win 65535, length 0
^C
9 packets captured
9 packets received by filter
0 packets dropped by kernel
so@so-virtual-machine:~$
```

TCP Handshake

- **[S]** SYN
- **[S.]** SYN/ACK
- **[.]** ACK



```
so@so-virtual-machine:~$ sudo tcpdump -ni eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
15:36:44.950939 IP 172.16.1.196.34779 > 172.16.1.2.53: 27501+ A? ad.samsclass.info. (35)
15:36:44.951234 IP 172.16.1.196.34779 > 172.16.1.2.53: 56778+ AAAA? ad.samsclass.info. (35)
15:36:44.955313 IP 172.16.1.2.53 > 172.16.1.196.34779: 27501 1/0/0 A 159.203.238.50 (51)
15:36:45.042196 IP 172.16.1.2.53 > 172.16.1.196.34779: 56778*- 2/0/0 CNAME ad.samsclass.info., A 159.203.238.50 (82)
15:36:45.042480 IP 172.16.1.196.48966 > 159.203.238.50.22: Flags [S], seq 3713351407, win 65535, options [mss 1460,sackOK,TS
 val 126922 ecr 0,nop,wscale 11], length 0
15:36:45.061954 IP 159.203.238.50.22 > 172.16.1.196.48966: Flags [S.], seq 3591988187, ack 3713351408, win 64240, options [m
ss 1460], length 0
15:36:45.061999 IP 172.16.1.196.48966 > 159.203.238.50.22: Flags [.], ack 1, win 65535, length 0
15:36:45.094071 IP 159.203.238.50.22 > 172.16.1.196.48966: Flags [P.], seq 1:42, ack 1, win 64240, length 41
15:36:45.094119 IP 172.16.1.196.48966 > 159.203.238.50.22: Flags [.], ack 42, win 65535, length 0
^C
9 packets captured
9 packets received by filter
0 packets dropped by kernel
so@so-virtual-machine:~$
```

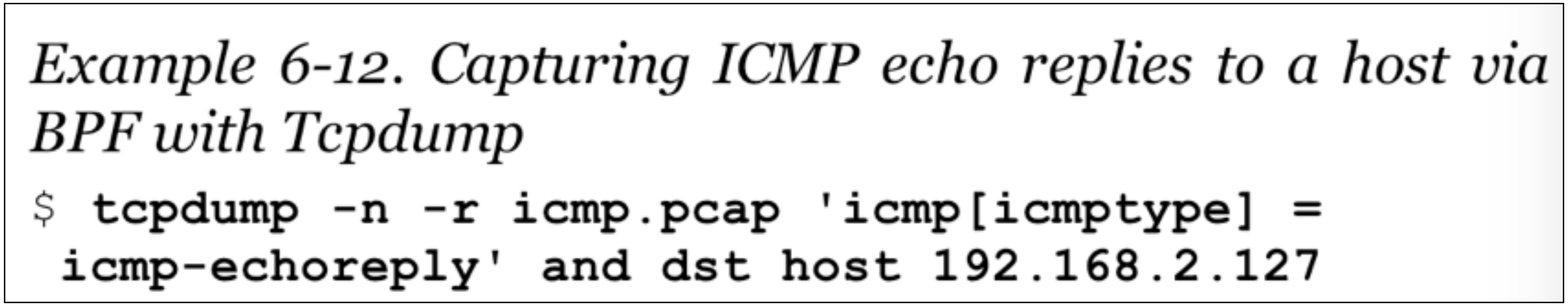
Capture Filters

- In Berkeley Packet Format (BPF)
- Add filter to the end of the command line
- **icmp** Only ICMP protocol
- **port 53** UDP or TCP port 53
- **tcp and port 443** Requires both conditions
- **man pcap-filter** to see all options

Capture Filters

- **host 192.168.1.1** traffic to or from this IP
- **src host 192.168.1.1** traffic from this IP
- **dst host 192.168.1.1** traffic to this IP
- **src net 192.168.1.0** traffic from this network

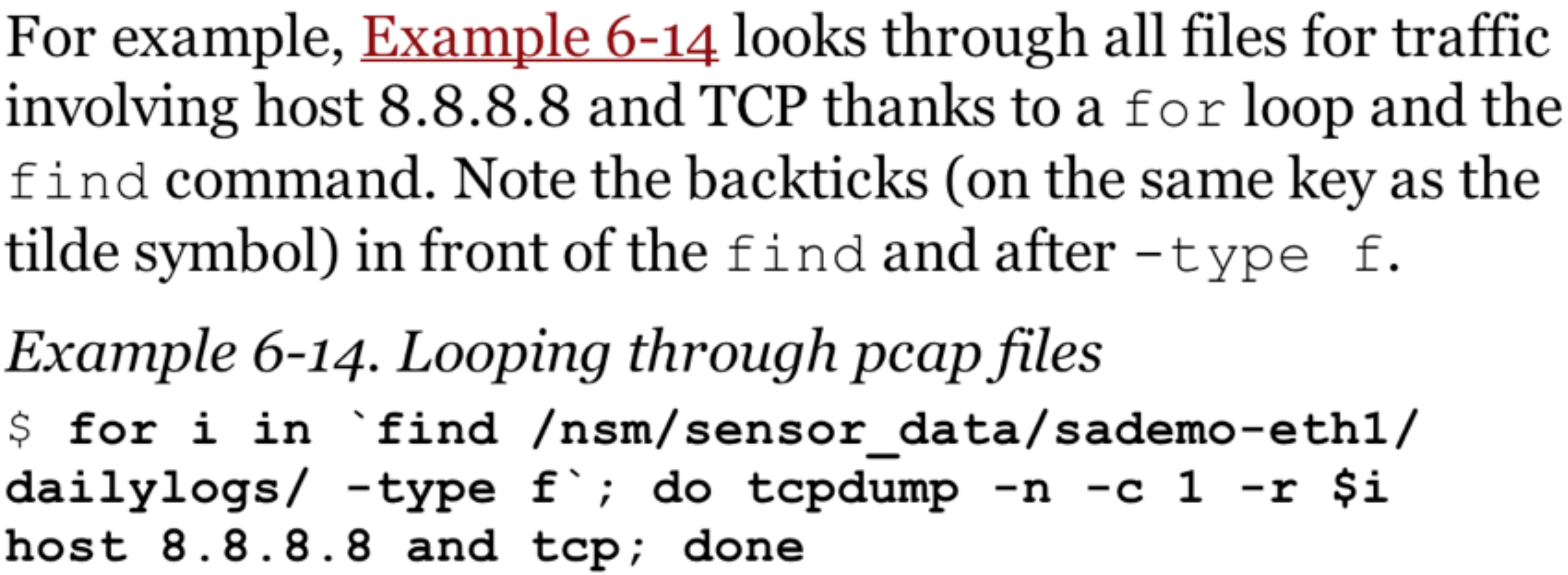
Only ICMP Replies



Example 6-12. Capturing ICMP echo replies to a host via BPF with Tcpdump

```
$ tcpdump -n -r icmp.pcap 'icmp[icmptype] =
 icmp-echoreply' and dst host 192.168.2.127
```

Looping Through Files



For example, Example 6-14 looks through all files for traffic involving host 8.8.8.8 and TCP thanks to a `for` loop and the `find` command. Note the backticks (on the same key as the tilde symbol) in front of the `find` and after `-type f`.

Example 6-14. Looping through pcap files

```
$ for i in `find /nsm/sensor_data/sademo-eth1/
dailylogs/ -type f`; do tcpdump -n -c 1 -r $i
host 8.8.8.8 and tcp; done
```

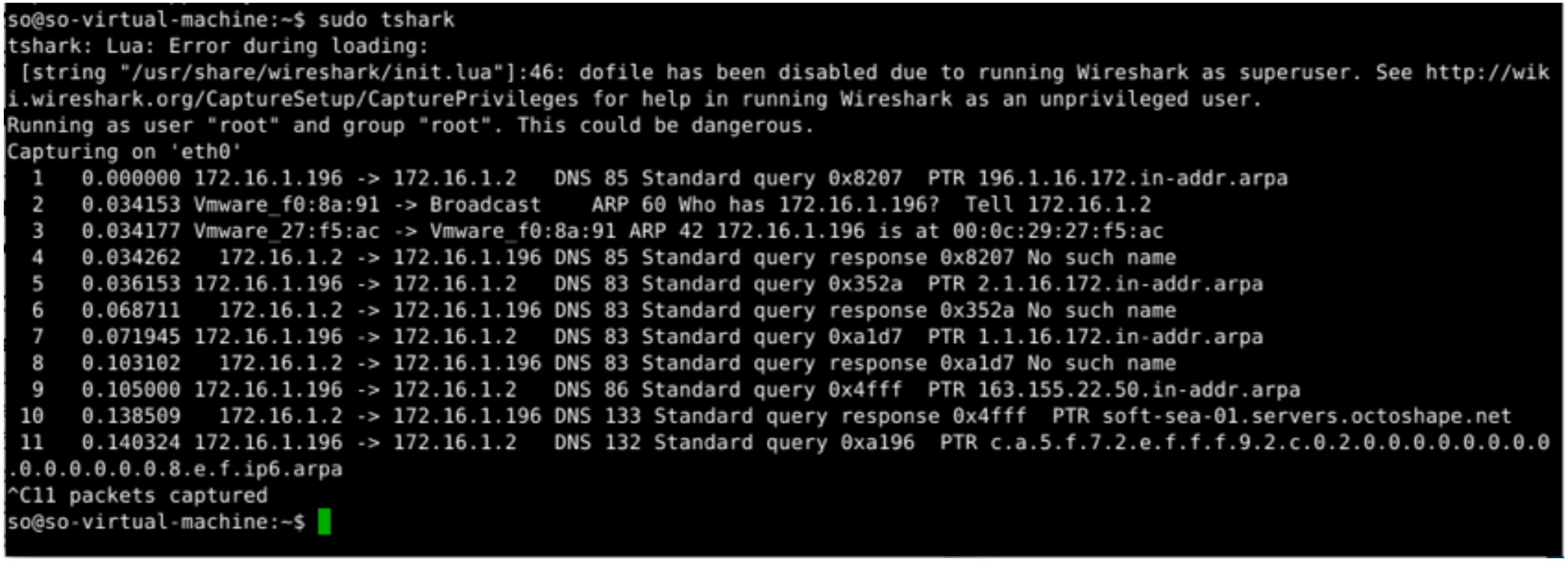
Kahoot!

Using Dumpcap and Tshark

Shipped with Wireshark

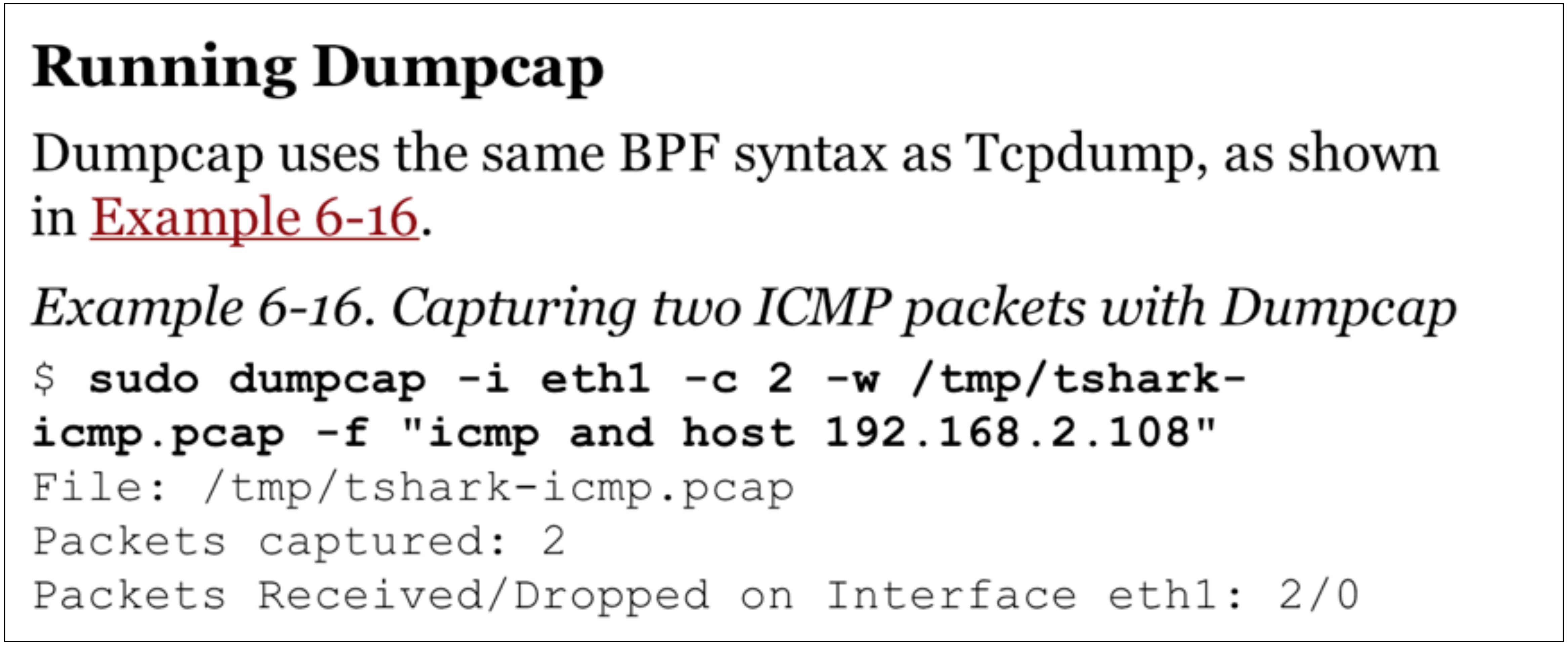
- Dumpcap is a simple packet collection tool
- Tshark is the command-line version of Wireshark
  - Analyzes traffic
  - Friendlier than tcpdump
  - Uses human-readable syntax

Tshark as Root



```
so@so-virtual-machine:~$ sudo tshark
tshark: Lua: Error during loading:
 [string "/usr/share/wireshark/init.lua"]:46: dofile has been disabled due to running Wireshark as superuser. See http://wik
i.wireshark.org/CaptureSetup/CapturePrivileges for help in running Wireshark as an unprivileged user.
Running as user "root" and group "root". This could be dangerous.
Capturing on 'eth0'
  1   0.000000 172.16.1.196 -> 172.16.1.2   DNS 85 Standard query 0x8207  PTR 196.1.16.172.in-addr.arpa
  2   0.034153 Vmware_f0:8a:91 -> Broadcast    ARP 60 Who has 172.16.1.196?  Tell 172.16.1.2
  3   0.034177 Vmware_27:f5:ac -> Vmware_f0:8a:91 ARP 42 172.16.1.196 is at 00:0c:29:27:f5:ac
  4   0.034262   172.16.1.2 -> 172.16.1.196 DNS 85 Standard query response 0x8207 No such name
  5   0.036153 172.16.1.196 -> 172.16.1.2   DNS 83 Standard query 0x352a  PTR 2.1.16.172.in-addr.arpa
  6   0.068711   172.16.1.2 -> 172.16.1.196 DNS 83 Standard query response 0x352a No such name
  7   0.071945 172.16.1.196 -> 172.16.1.2   DNS 83 Standard query 0xa1d7  PTR 1.1.16.172.in-addr.arpa
  8   0.103102   172.16.1.2 -> 172.16.1.196 DNS 83 Standard query response 0xa1d7 No such name
  9   0.105000 172.16.1.196 -> 172.16.1.2   DNS 86 Standard query 0x4fff  PTR 163.155.22.50.in-addr.arpa
 10   0.138509   172.16.1.2 -> 172.16.1.196 DNS 133 Standard query response 0x4fff  PTR soft-sea-01.servers.octoshape.net
 11   0.140324 172.16.1.196 -> 172.16.1.2   DNS 132 Standard query 0xa196  PTR c.a.5.f.7.2.e.f.f.f.9.2.c.0.2.0.0.0.0.0.0.0.0
.0.0.0.0.0.0.8.e.f.ip6.arpa
^C11 packets captured
so@so-virtual-machine:~$ 
```

- Protocol dissectors may contain vulnerabilities
- Recommended: collect with dumpcap, analyze later with tshark and wireshark



Running Dumpcap

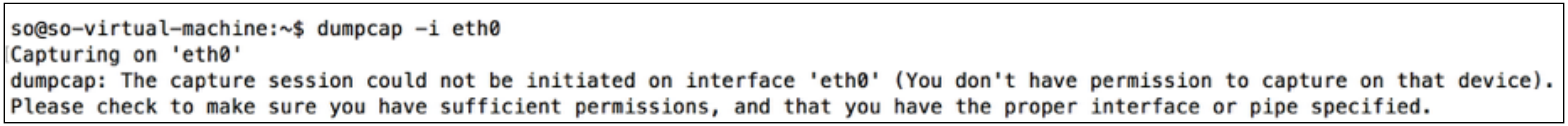
Dumpcap uses the same BPF syntax as Tcpdump, as shown in Example 6-16.

Example 6-16. Capturing two ICMP packets with Dumpcap

```
$ sudo dumpcap -i eth1 -c 2 -w /tmp/tshark-
icmp.pcap -f "icmp and host 192.168.2.108"
File: /tmp/tshark-icmp.pcap
Packets captured: 2
Packets Received/Dropped on Interface eth1: 2/0
```

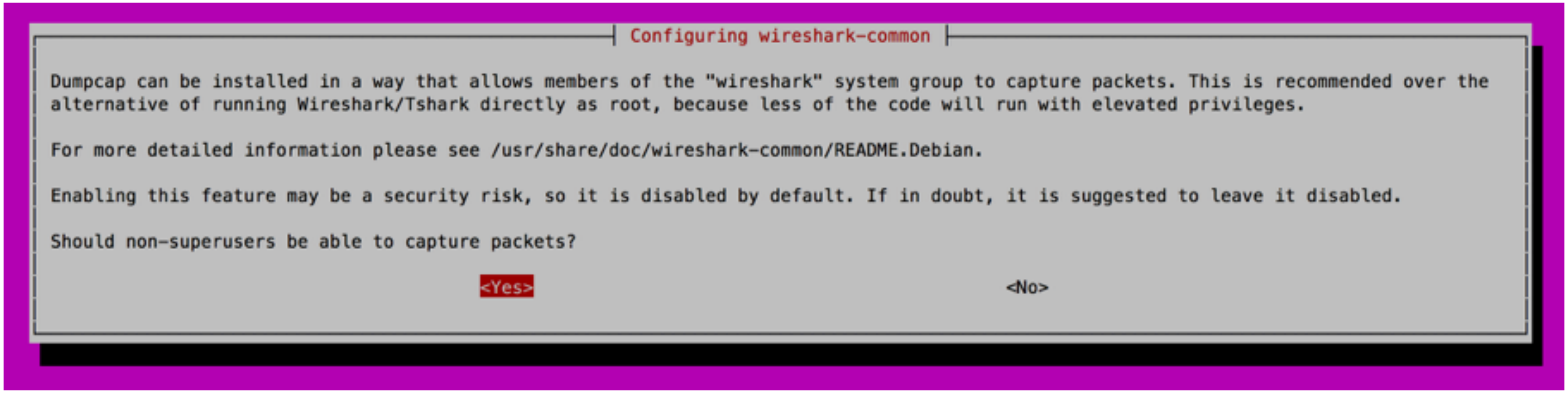
- When running as root, Dumpcap can't write to the user's home directory, so the output's in **/tmp**
- Dumpcap captures whole packets by default, unlike tcpdump

Running Dumpcap without root Privileges



```
so@so-virtual-machine:~$ dumpcap -i eth0
Capturing on 'eth0'
dumpcap: The capture session could not be initiated on interface 'eth0' (You don't have permission to capture on that device).
Please check to make sure you have sufficient permissions, and that you have the proper interface or pipe specified.
```

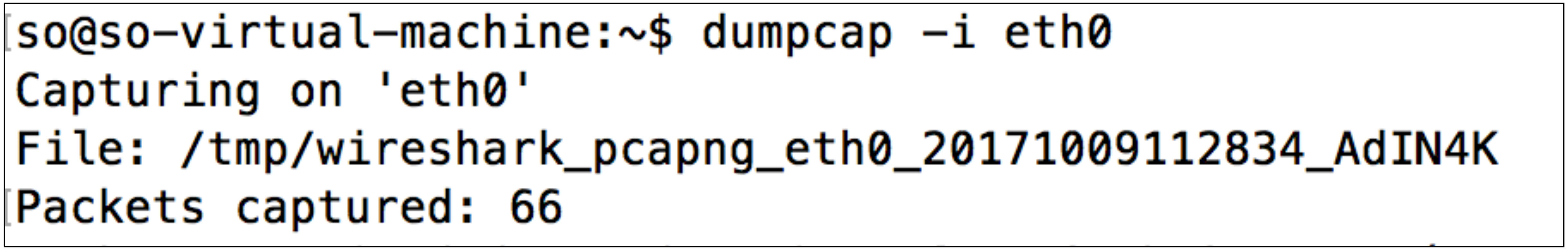
- **sudo dpkg-reconfigure wireshark-common**



```
Configuring wireshark-common

Dumpcap can be installed in a way that allows members of the "wireshark" system group to capture packets. This is recommended over the
alternative of running Wireshark/Tshark directly as root, because less of the code will run with elevated privileges.

For more detailed information please see /usr/share/doc/wireshark-common/README.Debian.

Enabling this feature may be a security risk, so it is disabled by default. If in doubt, it is suggested to leave it disabled.

Should non-superusers be able to capture packets?
```

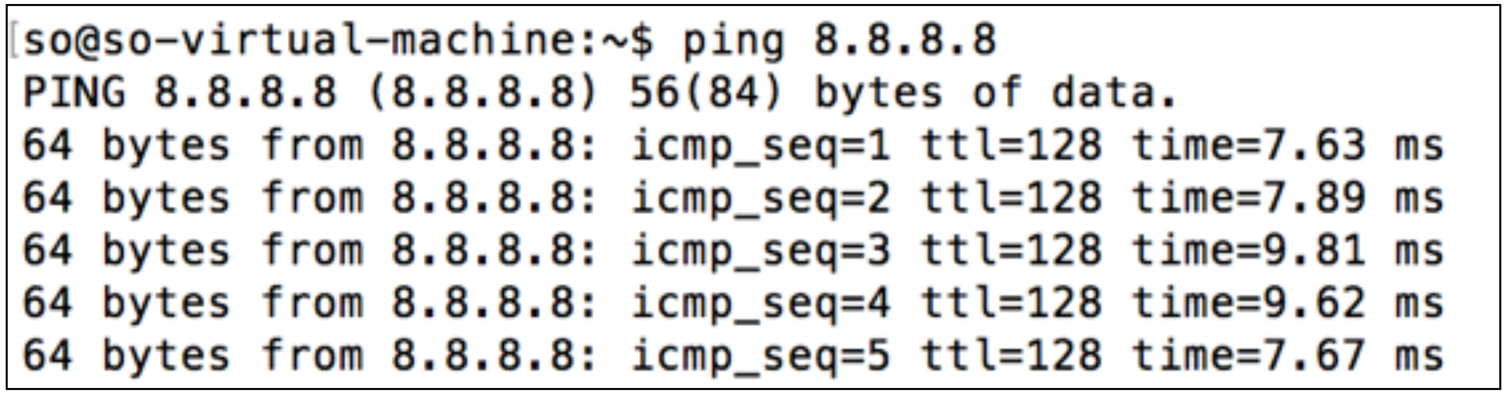
Running Dumpcap without root Privileges

- **sudo usermod -a -G wireshark so**
- **sudo reboot**

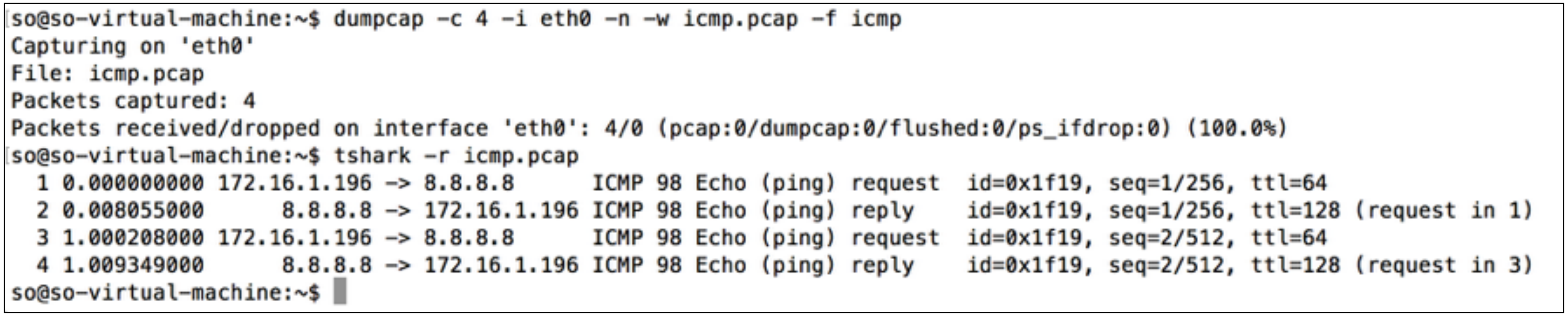


```
so@so-virtual-machine:~$ dumpcap -i eth0
Capturing on 'eth0'
File: /tmp/wireshark_pcapng_eth0_20171009112834_AdIN4K
Packets captured: 66
```

Capturing Pings with Dumpcap



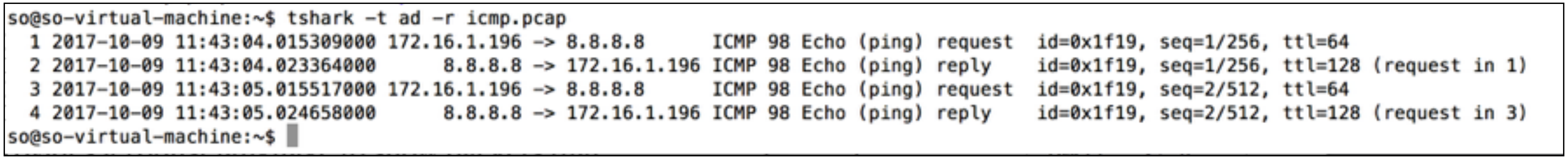
```
[so@so-virtual-machine:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=7.63 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=7.89 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=9.81 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=9.62 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=128 time=7.67 ms
```



```
[so@so-virtual-machine:~$ dumpcap -c 4 -i eth0 -n -w icmp.pcap -f icmp
Capturing on 'eth0'
File: icmp.pcap
Packets captured: 4
Packets received/dropped on interface 'eth0': 4/0 (pcap:0/dumpcap:0/flushed:0/ps_ifdrop:0) (100.0%)
[so@so-virtual-machine:~$ tshark -r icmp.pcap
  1 0.000000000 172.16.1.196 -> 8.8.8.8      ICMP 98 Echo (ping) request  id=0x1f19, seq=1/256, ttl=64
  2 0.008055000      8.8.8.8 -> 172.16.1.196 ICMP 98 Echo (ping) reply    id=0x1f19, seq=1/256, ttl=128 (request in 1)
  3 1.000208000 172.16.1.196 -> 8.8.8.8      ICMP 98 Echo (ping) request  id=0x1f19, seq=2/512, ttl=64
  4 1.009349000      8.8.8.8 -> 172.16.1.196 ICMP 98 Echo (ping) reply    id=0x1f19, seq=2/512, ttl=128 (request in 3)
so@so-virtual-machine:~$ █
```

Absolute Timestamps in Tshark

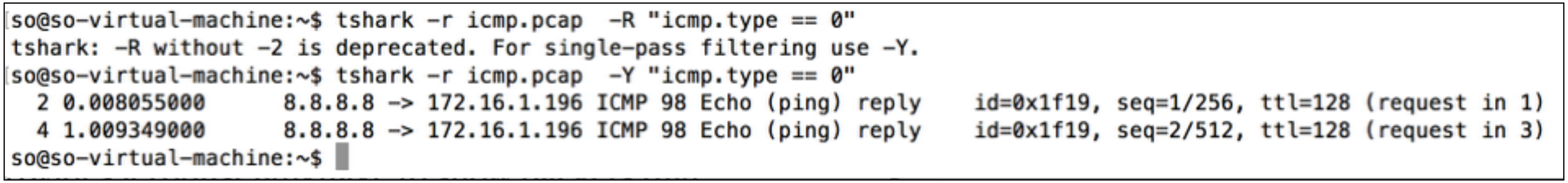
- **tshark -t ad -r icmp.pcap**



```
so@so-virtual-machine:~$ tshark -t ad -r icmp.pcap
  1 2017-10-09 11:43:04.015309000 172.16.1.196 -> 8.8.8.8      ICMP 98 Echo (ping) request  id=0x1f19, seq=1/256, ttl=64
  2 2017-10-09 11:43:04.023364000      8.8.8.8 -> 172.16.1.196 ICMP 98 Echo (ping) reply    id=0x1f19, seq=1/256, ttl=128 (request in 1)
  3 2017-10-09 11:43:05.015517000 172.16.1.196 -> 8.8.8.8      ICMP 98 Echo (ping) request  id=0x1f19, seq=2/512, ttl=64
  4 2017-10-09 11:43:05.024658000      8.8.8.8 -> 172.16.1.196 ICMP 98 Echo (ping) reply    id=0x1f19, seq=2/512, ttl=128 (request in 3)
so@so-virtual-machine:~$ 
```

Using Display Filters with Tshark

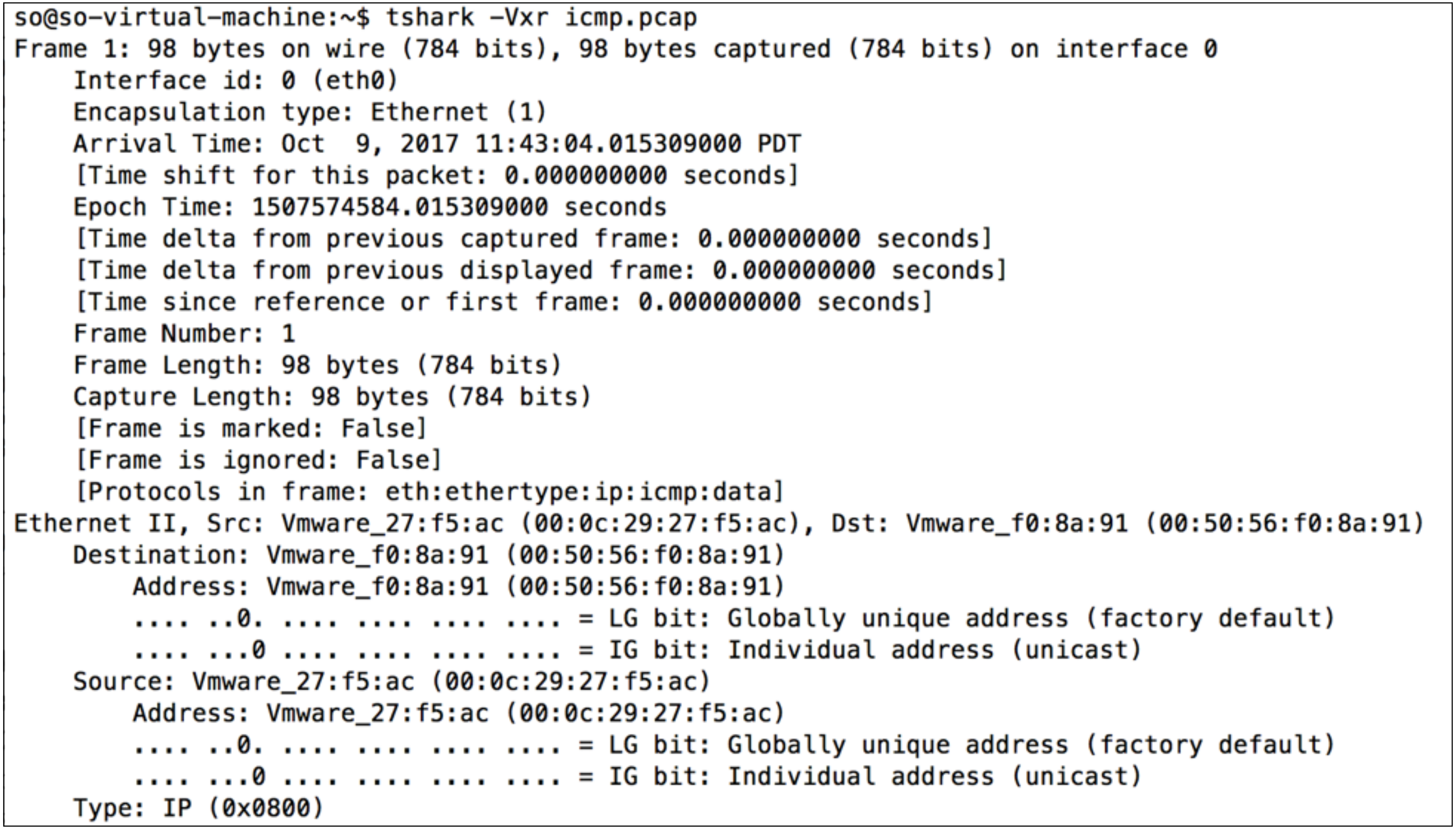
- Display filters use a different format than BPF
- Display filters don't affect packet capture
- **tshark -r icmp.pcap -Y "icmp.type == 0"**



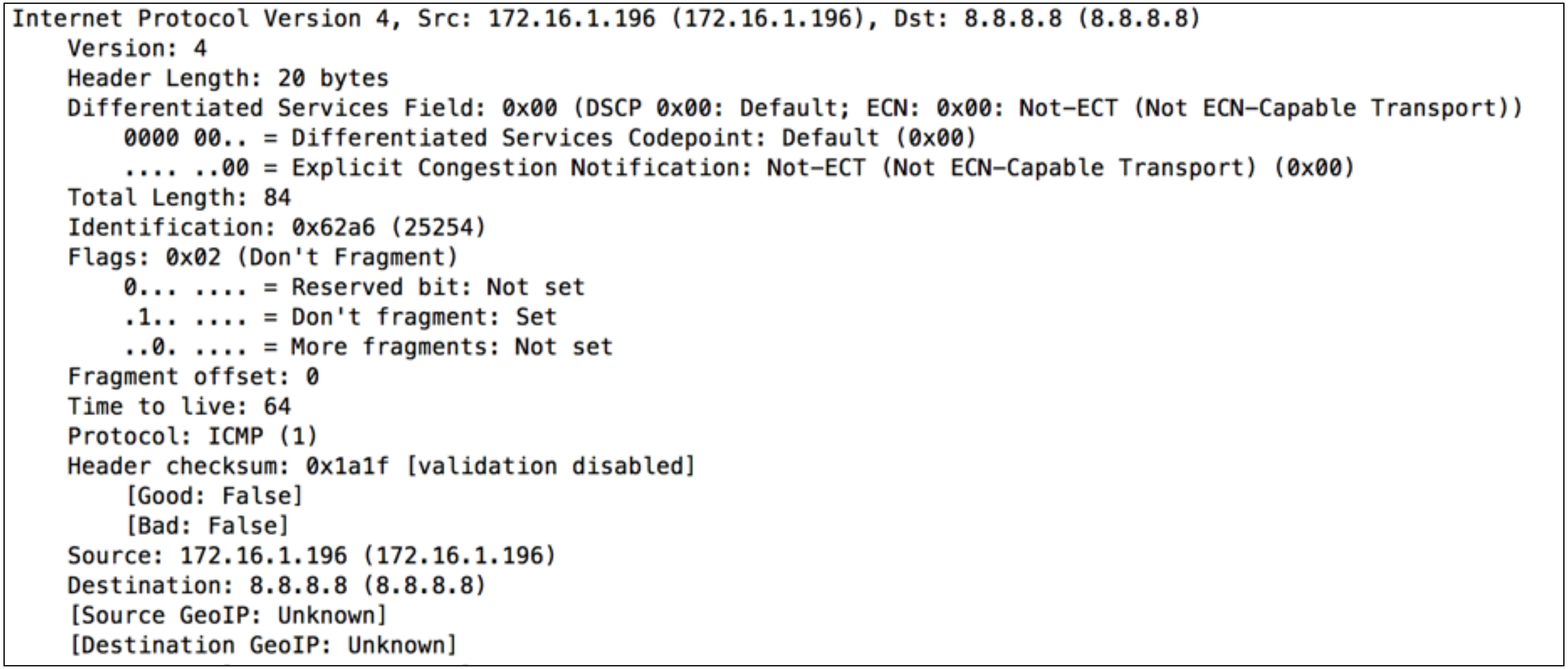
```
so@so-virtual-machine:~$ tshark -r icmp.pcap  -R "icmp.type == 0"
tshark: -R without -2 is deprecated. For single-pass filtering use -Y.
so@so-virtual-machine:~$ tshark -r icmp.pcap  -Y "icmp.type == 0"
  2 0.008055000      8.8.8.8 -> 172.16.1.196 ICMP 98 Echo (ping) reply    id=0x1f19, seq=1/256, ttl=128 (request in 1)
  4 1.009349000      8.8.8.8 -> 172.16.1.196 ICMP 98 Echo (ping) reply    id=0x1f19, seq=2/512, ttl=128 (request in 3)
so@so-virtual-machine:~$ 
```

Full Decode

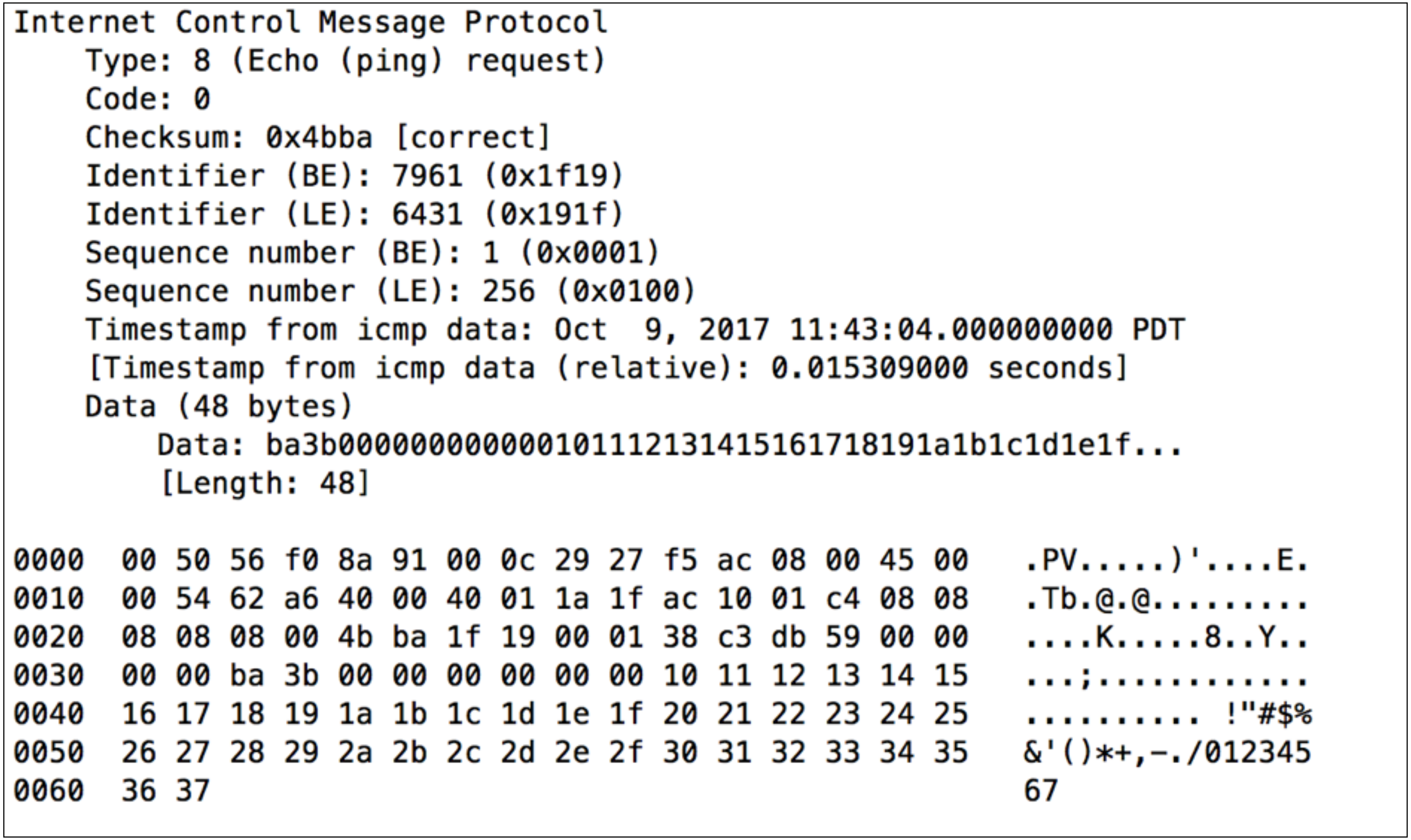
- **-V** for verbose protocol decode
- **-x** for hex and ASCII



```
so@so-virtual-machine:~$ tshark -Vxr icmp.pcap
Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
    Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Oct  9, 2017 11:43:04.015309000 PDT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1507574584.015309000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 98 bytes (784 bits)
    Capture Length: 98 bytes (784 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:icmp:data]
Ethernet II, Src: Vmware_27:f5:ac (00:0c:29:27:f5:ac), Dst: Vmware_f0:8a:91 (00:50:56:f0:8a:91)
    Destination: Vmware_f0:8a:91 (00:50:56:f0:8a:91)
        Address: Vmware_f0:8a:91 (00:50:56:f0:8a:91)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: Vmware_27:f5:ac (00:0c:29:27:f5:ac)
        Address: Vmware_27:f5:ac (00:0c:29:27:f5:ac)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
```



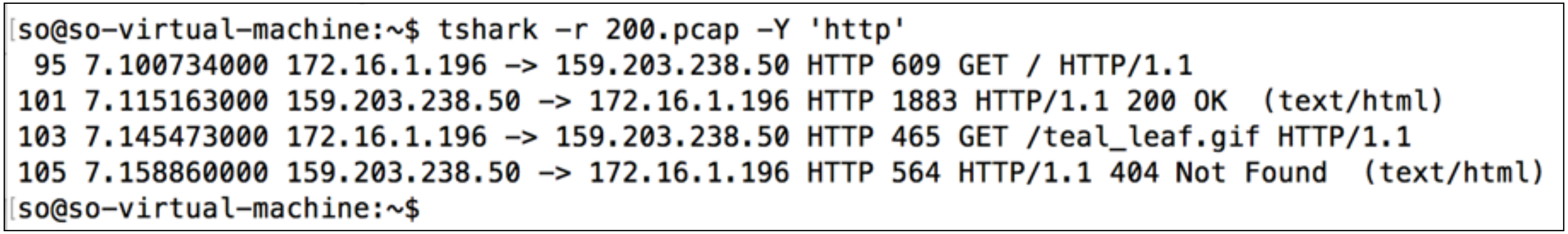
```
Internet Protocol Version 4, Src: 172.16.1.196 (172.16.1.196), Dst: 8.8.8.8 (8.8.8.8)
    Version: 4
    Header Length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)
    Total Length: 84
    Identification: 0x62a6 (25254)
    Flags: 0x02 (Don't Fragment)
        0... .... = Reserved bit: Not set
        .1.. .... = Don't fragment: Set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0x1a1f [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 172.16.1.196 (172.16.1.196)
    Destination: 8.8.8.8 (8.8.8.8)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
```



```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4bba [correct]
    Identifier (BE): 7961 (0x1f19)
    Identifier (LE): 6431 (0x191f)
    Sequence number (BE): 1 (0x0001)
    Sequence number (LE): 256 (0x0100)
    Timestamp from icmp data: Oct  9, 2017 11:43:04.000000000 PDT
    [Timestamp from icmp data (relative): 0.015309000 seconds]
    Data (48 bytes)
        Data: ba3b00000000000010111213141516171819 1a1b1c1d1e1f...
        [Length: 48]

0000  00 50 56 f0 8a 91 00 0c 29 27 f5 ac 08 00 45 00   .PV.....)'....E.
0010  00 54 62 a6 40 00 40 01 1a 1f ac 10 01 c4 08 08   .Tb.@.@.........
0020  08 08 08 00 4b ba 1f 19 00 01 38 c3 db 59 00 00   ....K.....8..Y..
0030  00 00 ba 3b 00 00 00 00 00 00 10 11 12 13 14 15   ...;............
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
0060  36 37                                             67
```

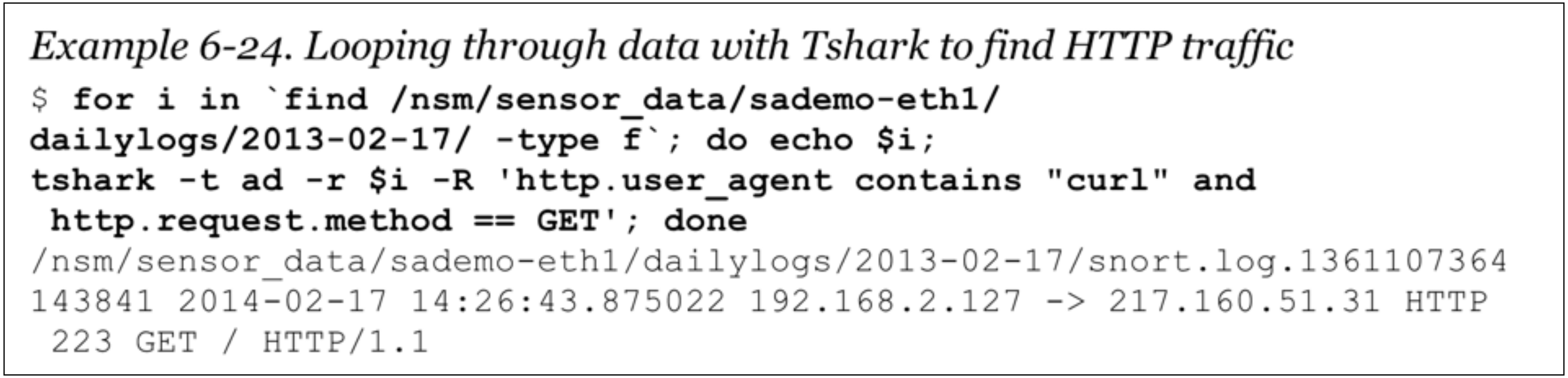
Tshark Display Filters in Action

- View HTTP Traffic



```
[so@so-virtual-machine:~$ tshark -r 200.pcap -Y 'http'
 95 7.100734000 172.16.1.196 -> 159.203.238.50 HTTP 609 GET / HTTP/1.1
101 7.115163000 159.203.238.50 -> 172.16.1.196 HTTP 1883 HTTP/1.1 200 OK  (text/html)
103 7.145473000 172.16.1.196 -> 159.203.238.50 HTTP 465 GET /teal_leaf.gif HTTP/1.1
105 7.158860000 159.203.238.50 -> 172.16.1.196 HTTP 564 HTTP/1.1 404 Not Found  (text/html)
[so@so-virtual-machine:~$
```

Tshark Display Filters in Action



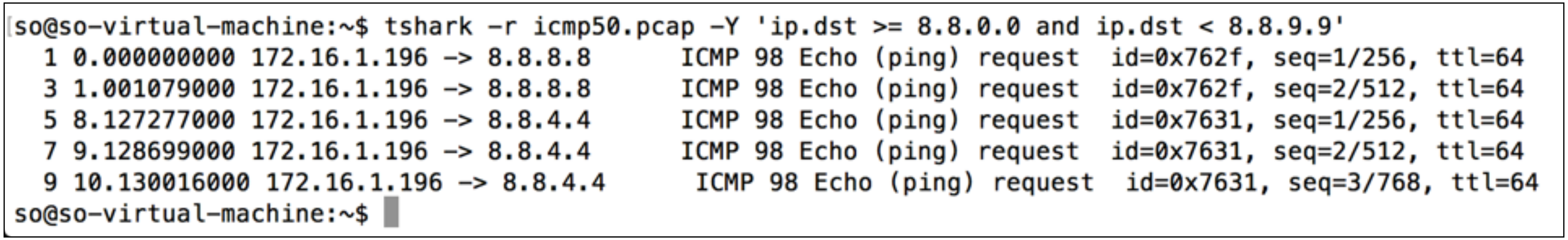
Example 6-24. Looping through data with Tshark to find HTTP traffic

```
$ for i in `find /nsm/sensor_data/sademo-eth1/
dailylogs/2013-02-17/ -type f`; do echo $i;
tshark -t ad -r $i -R 'http.user_agent contains "curl" and
 http.request.method == GET'; done
/nsm/sensor_data/sademo-eth1/dailylogs/2013-02-17/snort.log.1361107364
143841 2014-02-17 14:26:43.875022 192.168.2.127 -> 217.160.51.31 HTTP
 223 GET / HTTP/1.1
```

- Use **-Y** instead of **-R**

Tshark Display Filters in Action

- Searching for a range of IP addresses



```
so@so-virtual-machine:~$ tshark -r icmp50.pcap -Y 'ip.dst >= 8.8.0.0 and ip.dst < 8.8.9.9'
  1 0.000000000 172.16.1.196 -> 8.8.8.8      ICMP 98 Echo (ping) request  id=0x762f, seq=1/256, ttl=64
  3 1.001079000 172.16.1.196 -> 8.8.8.8      ICMP 98 Echo (ping) request  id=0x762f, seq=2/512, ttl=64
  5 8.127277000 172.16.1.196 -> 8.8.4.4      ICMP 98 Echo (ping) request  id=0x7631, seq=1/256, ttl=64
  7 9.128699000 172.16.1.196 -> 8.8.4.4      ICMP 98 Echo (ping) request  id=0x7631, seq=2/512, ttl=64
  9 10.130016000 172.16.1.196 -> 8.8.4.4      ICMP 98 Echo (ping) request  id=0x7631, seq=3/768, ttl=64
so@so-virtual-machine:~$ 
```

Running Argus and the Ra Client

Argus

- A session data generation and analysis suite

- Argus server is running by default on Security Onion

- Client is in **/nsm/sensor_data/<sensorname>/argus** directory

- **sudo nsm_sensor_ps-status --only-argus**

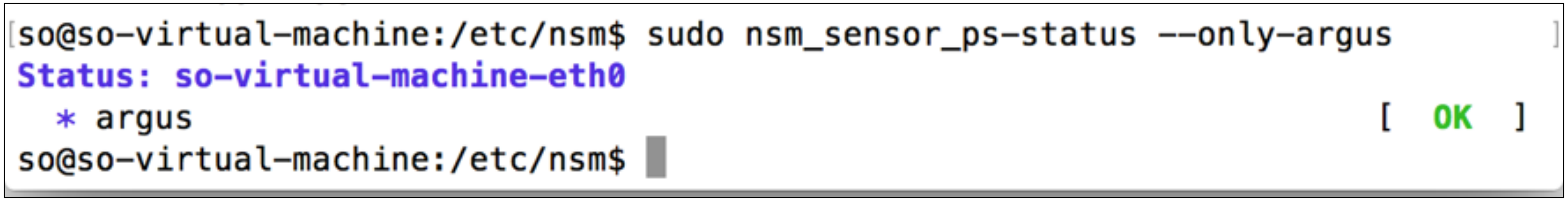
    - Shows Argus status

Was Off by Default

- Do this to start argus

- **sudo sed -i 's|ARGUS_ENABLED="no"| ARGUS_ENABLED="yes"|g' /etc/nsm/*/ sensor.conf**

- **sudo service nsm restart**

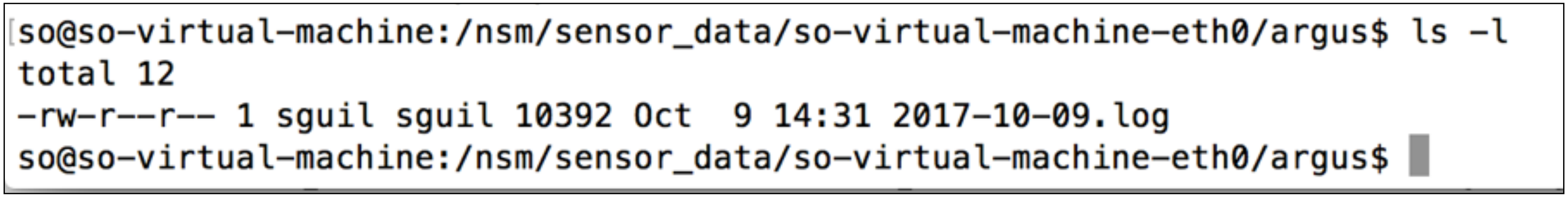


```
so@so-virtual-machine:/etc/nsm$ sudo nsm_sensor_ps-status --only-argus
Status: so-virtual-machine-eth0
  * argus                                                          [  OK  ]
so@so-virtual-machine:/etc/nsm$ 
```

Stopping and Starting Argus

- **sudo nsm_sensor_ps-stop --only-argus**
- **sudo nsm_sensor_ps-start --only-argus**

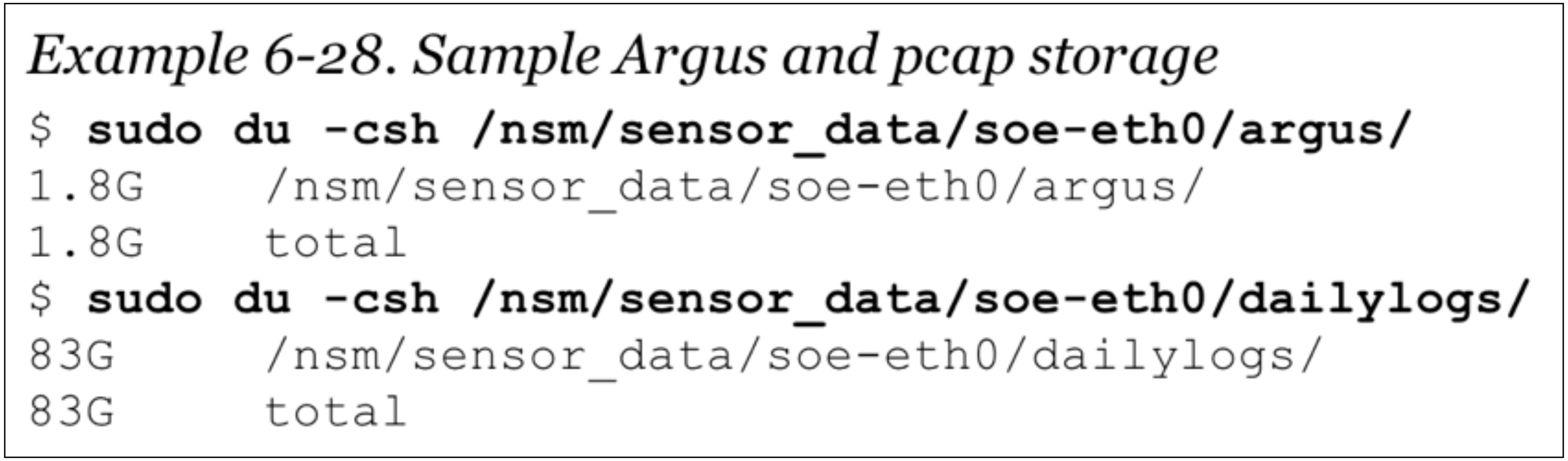
Argus Data



```
so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ ls -l
total 12
-rw-r--r-- 1 sguil sguil 10392 Oct  9 14:31 2017-10-09.log
so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ 
```

Argus File Format

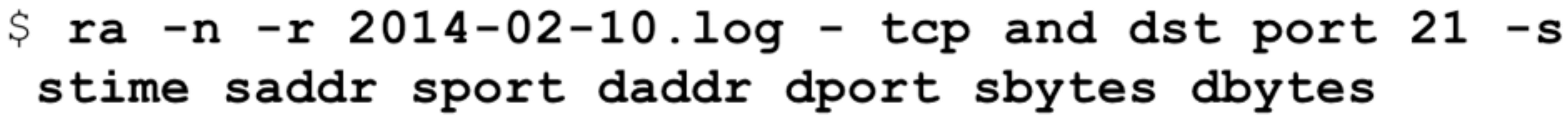
- Argus stores flows, not complete pcaps
- Much smaller: ex: 48 days of data



Example 6-28. Sample Argus and pcap storage

```
$ sudo du -csh /nsm/sensor_data/soe-eth0/argus/
1.8G    /nsm/sensor_data/soe-eth0/argus/
1.8G    total
$ sudo du -csh /nsm/sensor_data/soe-eth0/dailylogs/
83G     /nsm/sensor_data/soe-eth0/dailylogs/
83G     total
```

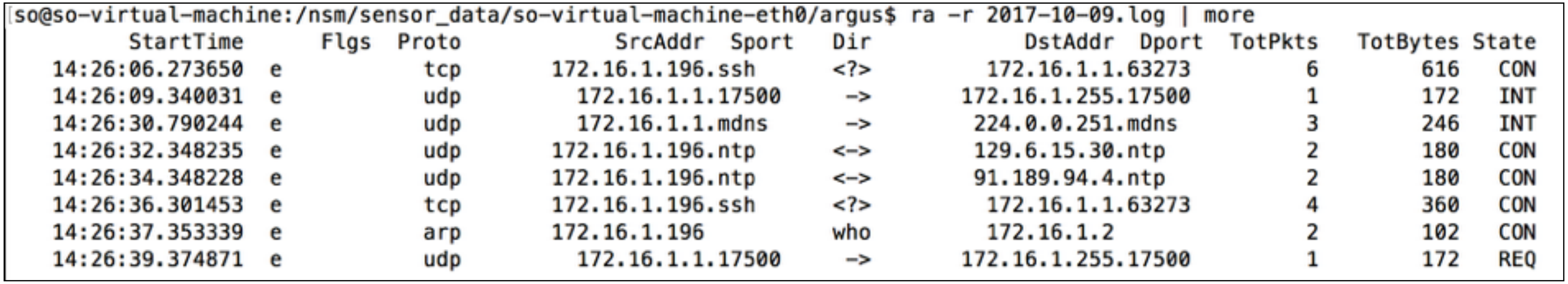
Examining Argus Data



```
$ ra -n -r 2014-02-10.log - tcp and dst port 21 -s
 stime saddr sport daddr dport sbytes dbytes
```

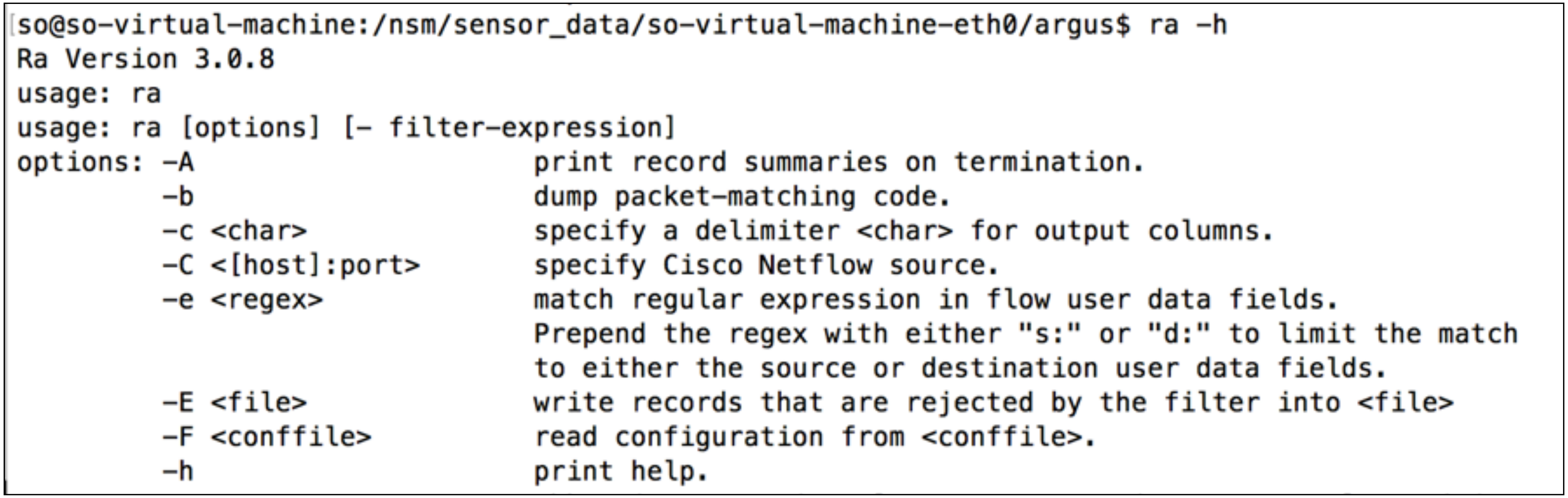
- **-n** Don't resolve port numbers to names
- **tcp and dst port 21** BPF packet filter
- **-s** Specify which fields to display

Argus Data in SO



```
[so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ ra -r 2017-10-09.log | more
         StartTime      Flgs  Proto            SrcAddr   Sport   Dir            DstAddr  Dport  TotPkts   TotBytes State
   14:26:06.273650  e           tcp       172.16.1.196.ssh       <?>         172.16.1.1.63273         6        616   CON
   14:26:09.340031  e           udp         172.16.1.1.17500      ->       172.16.1.255.17500         1        172   INT
   14:26:30.790244  e           udp         172.16.1.1.mdns       ->        224.0.0.251.mdns          3        246   INT
   14:26:32.348235  e           udp       172.16.1.196.ntp       <->        129.6.15.30.ntp           2        180   CON
   14:26:34.348228  e           udp       172.16.1.196.ntp       <->        91.189.94.4.ntp           2        180   CON
   14:26:36.301453  e           tcp       172.16.1.196.ssh       <?>         172.16.1.1.63273         4        360   CON
   14:26:37.353339  e           arp       172.16.1.196           who         172.16.1.2               2        102   CON
   14:26:39.374871  e           udp         172.16.1.1.17500      ->       172.16.1.255.17500         1        172   REQ
```

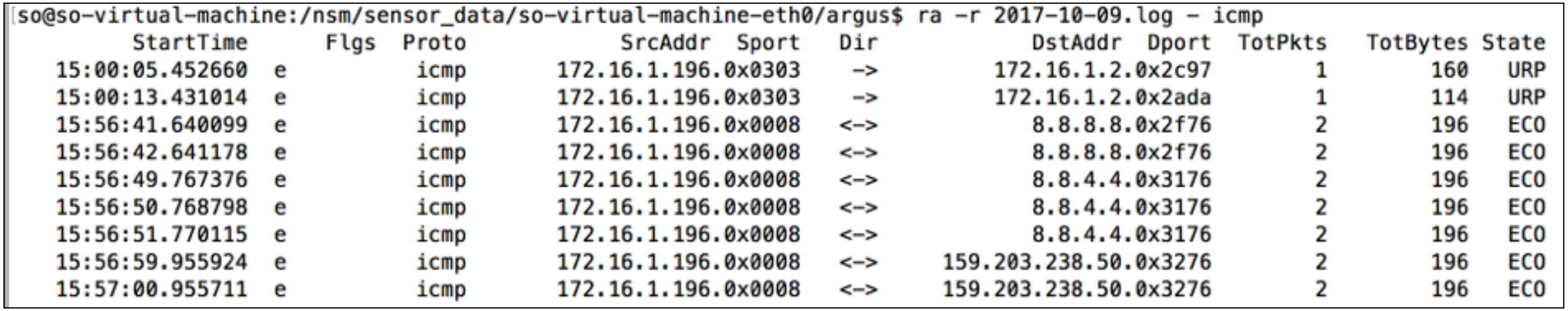
Ra Help



```
so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ ra -h
Ra Version 3.0.8
usage: ra
usage: ra [options] [- filter-expression]
options: -A                       print record summaries on termination.
         -b                       dump packet-matching code.
         -c <char>                specify a delimiter <char> for output columns.
         -C <[host]:port>         specify Cisco Netflow source.
         -e <regex>               match regular expression in flow user data fields.
                                  Prepend the regex with either "s:" or "d:" to limit the match
                                  to either the source or destination user data fields.
         -E <file>                write records that are rejected by the filter into <file>
         -F <conffile>            read configuration from <conffile>.
         -h                       print help.
```

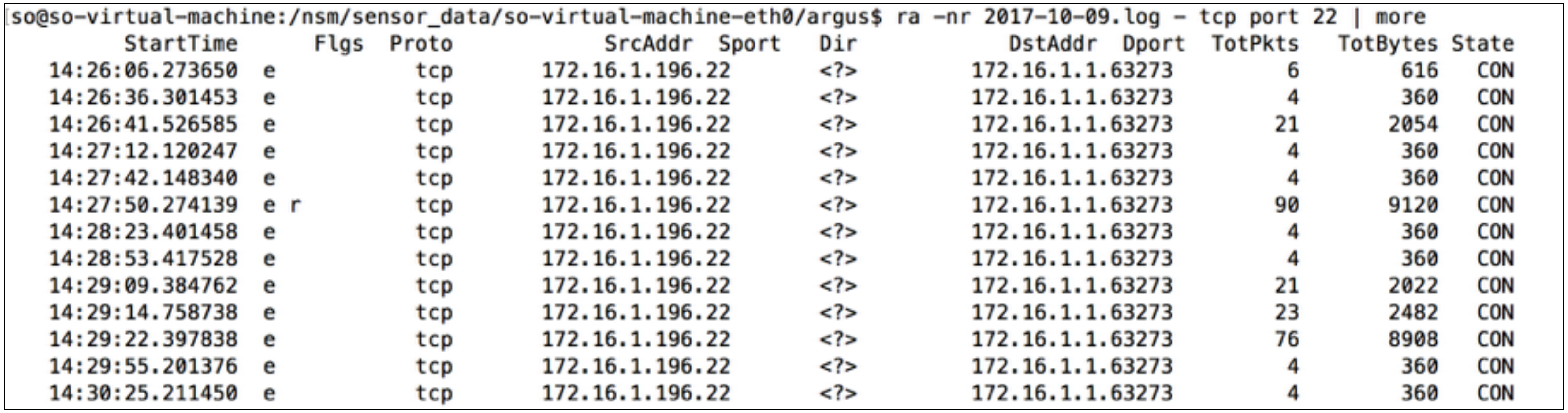
- **-** switch to filter

Ra Filtered for ICMP



```
[so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ ra -r 2017-10-09.log - icmp
        StartTime      Flgs  Proto          SrcAddr  Sport   Dir          DstAddr  Dport  TotPkts   TotBytes State
   15:00:05.452660  e          icmp   172.16.1.196.0x0303    ->       172.16.1.2.0x2c97        1        160   URP
   15:00:13.431014  e          icmp   172.16.1.196.0x0303    ->       172.16.1.2.0x2ada        1        114   URP
   15:56:41.640099  e          icmp   172.16.1.196.0x0008   <->          8.8.8.8.0x2f76        2        196   ECO
   15:56:42.641178  e          icmp   172.16.1.196.0x0008   <->          8.8.8.8.0x2f76        2        196   ECO
   15:56:49.767376  e          icmp   172.16.1.196.0x0008   <->          8.8.4.4.0x3176        2        196   ECO
   15:56:50.768798  e          icmp   172.16.1.196.0x0008   <->          8.8.4.4.0x3176        2        196   ECO
   15:56:51.770115  e          icmp   172.16.1.196.0x0008   <->          8.8.4.4.0x3176        2        196   ECO
   15:56:59.955924  e          icmp   172.16.1.196.0x0008   <->   159.203.238.50.0x3276        2        196   ECO
   15:57:00.955711  e          icmp   172.16.1.196.0x0008   <->   159.203.238.50.0x3276        2        196   ECO
```

Ra for SSH

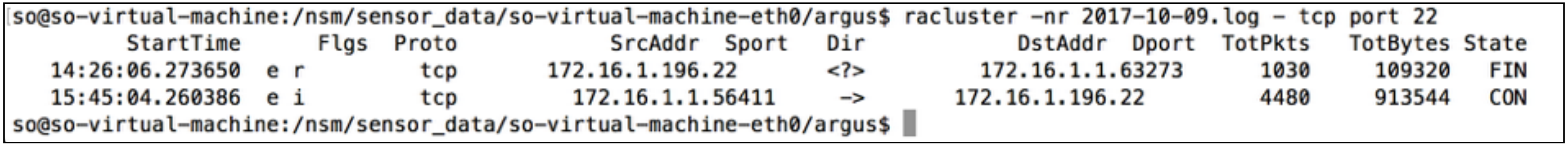


```
[so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ ra -nr 2017-10-09.log - tcp port 22 | more
         StartTime      Flgs  Proto            SrcAddr  Sport   Dir            DstAddr  Dport  TotPkts   TotBytes State
   14:26:06.273650  e           tcp       172.16.1.196.22        <?>        172.16.1.1.63273         6        616   CON
   14:26:36.301453  e           tcp       172.16.1.196.22        <?>        172.16.1.1.63273         4        360   CON
   14:26:41.526585  e           tcp       172.16.1.196.22        <?>        172.16.1.1.63273        21       2054   CON
   14:27:12.120247  e           tcp       172.16.1.196.22        <?>        172.16.1.1.63273         4        360   CON
   14:27:42.148340  e           tcp       172.16.1.196.22        <?>        172.16.1.1.63273         4        360   CON
   14:27:50.274139  e r         tcp       172.16.1.196.22        <?>        172.16.1.1.63273        90       9120   CON
   14:28:23.401458  e           tcp       172.16.1.196.22        <?>        172.16.1.1.63273         4        360   CON
   14:28:53.417528  e           tcp       172.16.1.196.22        <?>        172.16.1.1.63273         4        360   CON
   14:29:09.384762  e           tcp       172.16.1.196.22        <?>        172.16.1.1.63273        21       2022   CON
   14:29:14.758738  e           tcp       172.16.1.196.22        <?>        172.16.1.1.63273        23       2482   CON
   14:29:22.397838  e           tcp       172.16.1.196.22        <?>        172.16.1.1.63273        76       8908   CON
   14:29:55.201376  e           tcp       172.16.1.196.22        <?>        172.16.1.1.63273         4        360   CON
   14:30:25.211450  e           tcp       172.16.1.196.22        <?>        172.16.1.1.63273         4        360   CON
```

- Many records for the same conversation

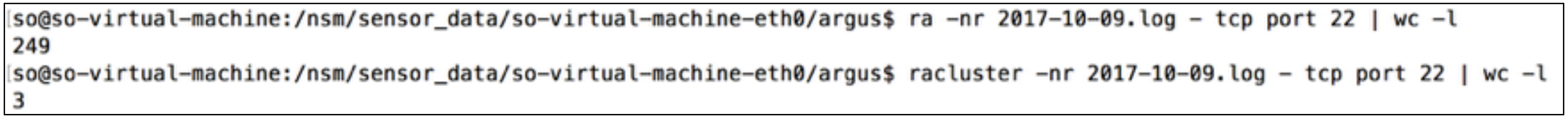
Racluster

- Ra can break a long conversation into separate sections
- Racluster combines them into one record



```
[so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ racluster -nr 2017-10-09.log - tcp port 22
         StartTime      Flgs  Proto            SrcAddr  Sport   Dir         DstAddr  Dport  TotPkts   TotBytes State
   14:26:06.273650  e r         tcp      172.16.1.196.22        <?>       172.16.1.1.63273     1030     109320   FIN
   15:45:04.260386  e i         tcp        172.16.1.1.56411      ->      172.16.1.196.22         4480     913544   CON
so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ 
```

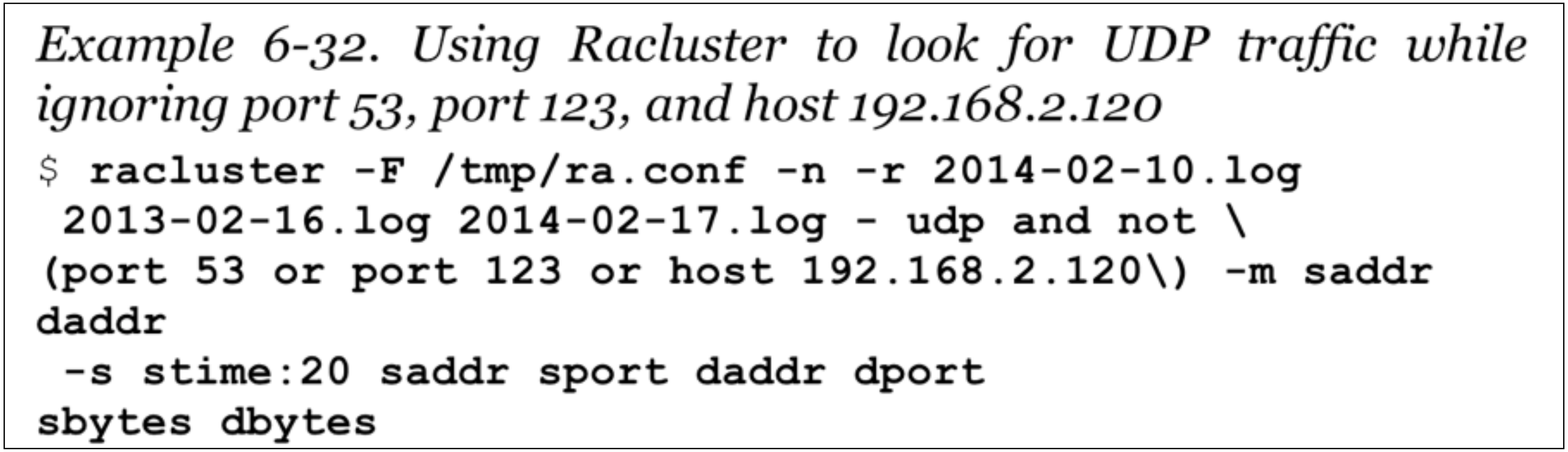
Number of Lines



```
[so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ ra -nr 2017-10-09.log - tcp port 22 | wc -l
249
[so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ racluster -nr 2017-10-09.log - tcp port 22 | wc -l
3
```

Advanced Usage Example

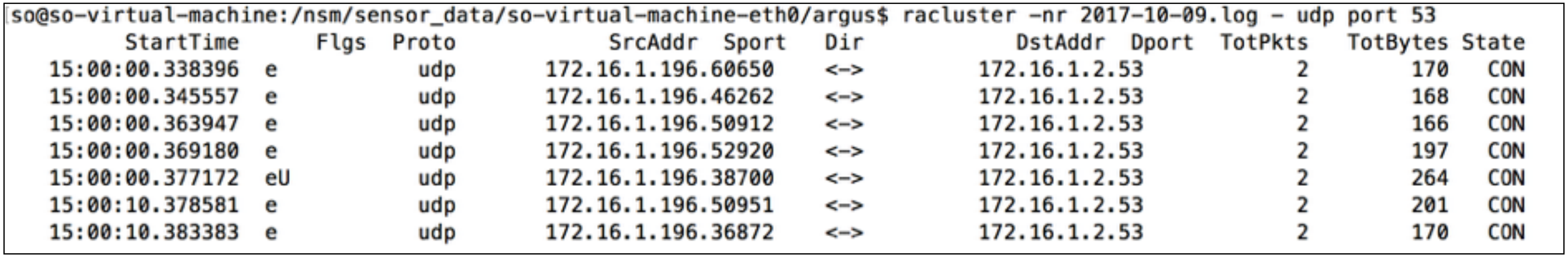
- **-m saddr daddr** groups records by source and destination IP address



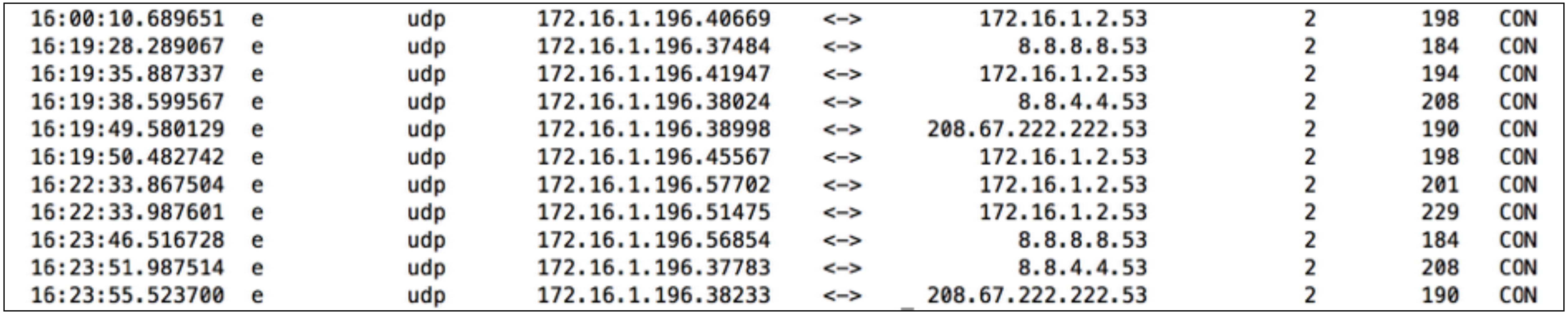
Example 6-32. Using Racluster to look for UDP traffic while ignoring port 53, port 123, and host 192.168.2.120

```
$ racluster -F /tmp/ra.conf -n -r 2014-02-10.log
 2013-02-16.log 2014-02-17.log - udp and not \
(port 53 or port 123 or host 192.168.2.120\) -m saddr
daddr
 -s stime:20 saddr sport daddr dport
sbytes dbytes
```

Without **-m**



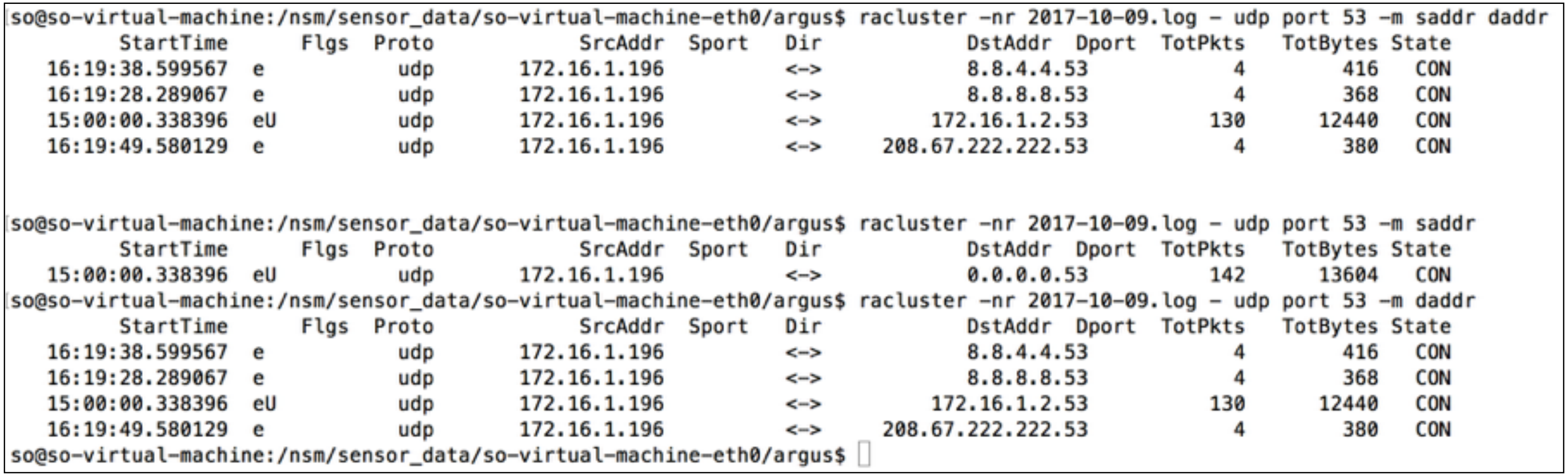
```
[so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ racluster -nr 2017-10-09.log - udp port 53
        StartTime      Flgs  Proto            SrcAddr  Sport   Dir         DstAddr  Dport  TotPkts   TotBytes State
   15:00:00.338396  e          udp       172.16.1.196.60650    <->       172.16.1.2.53           2        170   CON
   15:00:00.345557  e          udp       172.16.1.196.46262    <->       172.16.1.2.53           2        168   CON
   15:00:00.363947  e          udp       172.16.1.196.50912    <->       172.16.1.2.53           2        166   CON
   15:00:00.369180  e          udp       172.16.1.196.52920    <->       172.16.1.2.53           2        197   CON
   15:00:00.377172  eU         udp       172.16.1.196.38700    <->       172.16.1.2.53           2        264   CON
   15:00:10.378581  e          udp       172.16.1.196.50951    <->       172.16.1.2.53           2        201   CON
   15:00:10.383383  e          udp       172.16.1.196.36872    <->       172.16.1.2.53           2        170   CON
```



```
 16:00:10.689651  e          udp       172.16.1.196.40669    <->       172.16.1.2.53           2        198   CON
 16:19:28.289067  e          udp       172.16.1.196.37484    <->           8.8.8.8.53          2        184   CON
 16:19:35.887337  e          udp       172.16.1.196.41947    <->       172.16.1.2.53           2        194   CON
 16:19:38.599567  e          udp       172.16.1.196.38024    <->           8.8.4.4.53          2        208   CON
 16:19:49.580129  e          udp       172.16.1.196.38998    <->    208.67.222.222.53          2        190   CON
 16:19:50.482742  e          udp       172.16.1.196.45567    <->       172.16.1.2.53           2        198   CON
 16:22:33.867504  e          udp       172.16.1.196.57702    <->       172.16.1.2.53           2        201   CON
 16:22:33.987601  e          udp       172.16.1.196.51475    <->       172.16.1.2.53           2        229   CON
 16:23:46.516728  e          udp       172.16.1.196.56854    <->           8.8.8.8.53          2        184   CON
 16:23:51.987514  e          udp       172.16.1.196.37783    <->           8.8.4.4.53          2        208   CON
 16:23:55.523700  e          udp       172.16.1.196.38233    <->  _ 208.67.222.222.53          2        190   CON
```

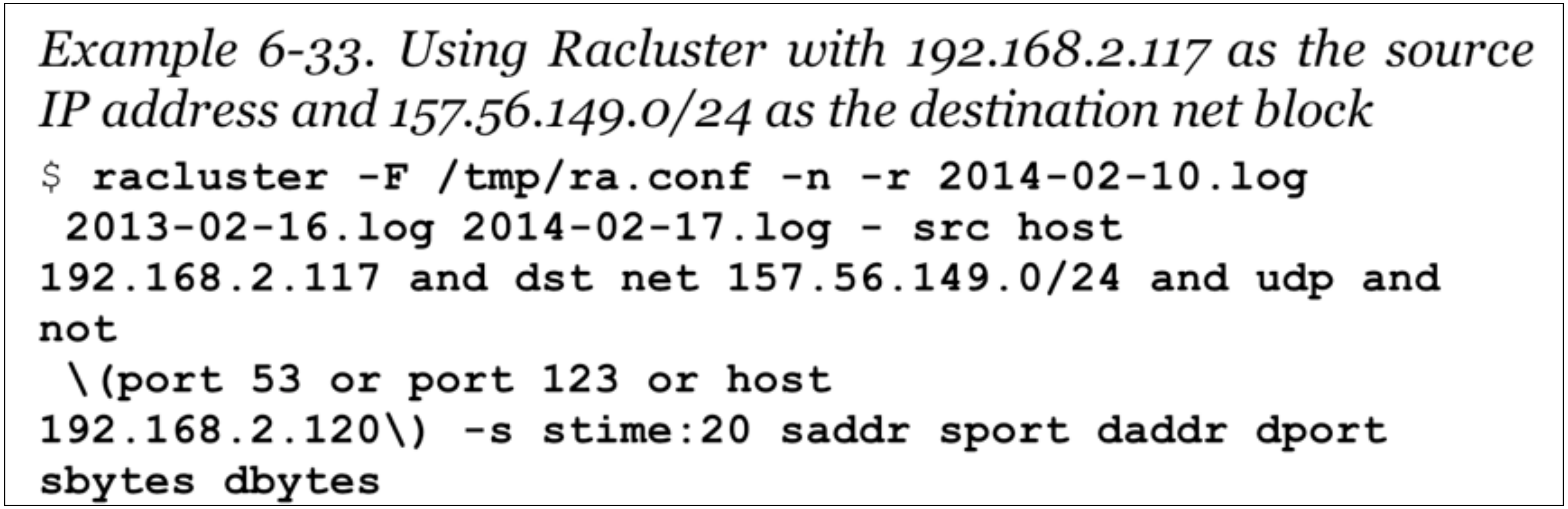
With **-m**

- Combines many conversations into one record



```
[so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ racluster -nr 2017-10-09.log - udp port 53 -m saddr daddr
        StartTime      Flgs  Proto            SrcAddr  Sport   Dir             DstAddr  Dport  TotPkts    TotBytes State
   16:19:38.599567  e          udp       172.16.1.196          <->           8.8.4.4.53              4         416   CON
   16:19:28.289067  e          udp       172.16.1.196          <->           8.8.8.8.53              4         368   CON
   15:00:00.338396  eU         udp       172.16.1.196          <->        172.16.1.2.53            130       12440   CON
   16:19:49.580129  e          udp       172.16.1.196          <->     208.67.222.222.53             4         380   CON


[so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ racluster -nr 2017-10-09.log - udp port 53 -m saddr
        StartTime      Flgs  Proto            SrcAddr  Sport   Dir             DstAddr  Dport  TotPkts    TotBytes State
   15:00:00.338396  eU         udp       172.16.1.196          <->           0.0.0.0.53            142       13604   CON
[so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ racluster -nr 2017-10-09.log - udp port 53 -m daddr
        StartTime      Flgs  Proto            SrcAddr  Sport   Dir             DstAddr  Dport  TotPkts    TotBytes State
   16:19:38.599567  e          udp       172.16.1.196          <->           8.8.4.4.53              4         416   CON
   16:19:28.289067  e          udp       172.16.1.196          <->           8.8.8.8.53              4         368   CON
   15:00:00.338396  eU         udp       172.16.1.196          <->        172.16.1.2.53            130       12440   CON
   16:19:49.580129  e          udp       172.16.1.196          <->     208.67.222.222.53             4         380   CON
so@so-virtual-machine:/nsm/sensor_data/so-virtual-machine-eth0/argus$ 
```

Advanced Usage Example



Example 6-33. Using Racluster with 192.168.2.117 as the source IP address and 157.56.149.0/24 as the destination net block

```
$ racluster -F /tmp/ra.conf -n -r 2014-02-10.log
 2013-02-16.log 2014-02-17.log - src host
192.168.2.117 and dst net 157.56.149.0/24 and udp and
not
 \(port 53 or port 123 or host
192.168.2.120\) -s stime:20 saddr sport daddr dport
sbytes dbytes
```

Kahoot!