

# Capturing network traffic

An introduction to Ethereal and Tcpdump

Jon-Olov Vatn

# Network sniffers

- Purpose of using them
  - Educational
  - Troubleshooting
  - Network measurements and analysis
  - Eavesdropping data communication
- Common tools
  - Ethereal (graphical), <http://www.ethereal.com>
  - Tcpdump (console), <http://www.tcpdump.org>

# Ethereal (graphical)

Packet list pane

Tree view pane

Byte view pane

The screenshot displays the Ethereal network protocol analyzer interface. The window title is 'ping-gw.pcap - Ethereal'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, and Help. The toolbar contains various icons for file operations and analysis. A filter bar is present with a dropdown menu and buttons for 'Expression...', 'Clear', and 'Apply'. The main area is divided into three panes:

- Packet list pane:** A table showing a list of captured packets. The second packet is selected.
- Tree view pane:** A hierarchical view of the selected packet's structure, showing Ethernet II, IP, and ICMP layers.
- Byte view pane:** A hexadecimal and ASCII representation of the selected packet's raw bytes.

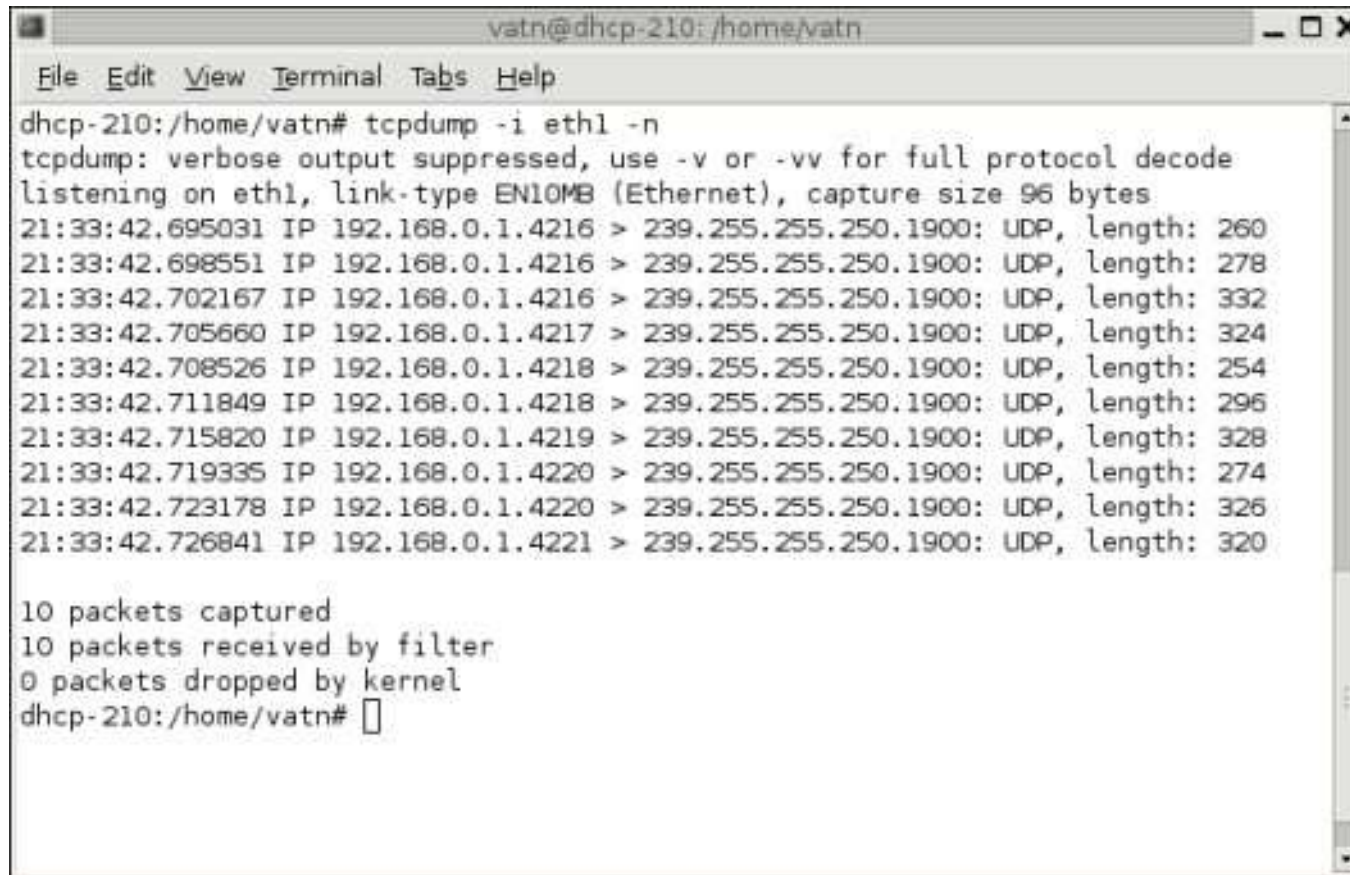
No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.10	Broadcast	ARP	Who has 192.168.0.1? Tell 192.168.0.10
2	0.002564	192.168.0.1	192.168.0.10	ARP	192.168.0.1 is at 00:0d:88:20:bb:7a
3	0.002598	192.168.0.100	192.168.0.1	ICMP	Echo (ping) request
4	0.004199	192.168.0.1	192.168.0.100	ICMP	Echo (ping) reply
5	1.000956	192.168.0.100	192.168.0.1	ICMP	Echo (ping) request
6	1.002560	192.168.0.1	192.168.0.100	ICMP	Echo (ping) reply
7	2.001802	192.168.0.100	192.168.0.1	ICMP	Echo (ping) request
8	2.003407	192.168.0.1	192.168.0.100	ICMP	Echo (ping) reply

Hardware type: Ethernet (0x0001)  
Protocol type: IP (0x0800)  
Hardware size: 6  
Protocol size: 4  
Opcode: reply (0x0002)  
Sender MAC address: 00:0d:88:20:bb:7a (192.168.0.1)

```
0000 00 02 8a 78 90 c5 00 0d 88 20 bb 7a 08 06 00 01  ...X... .Z...
0010 08 00 06 04 00 02 00 0d 88 20 bb 7a c0 a8 00 01  ..... .Z...
0020 00 02 8a 78 90 c5 c0 a8 00 64 34 81 04 08 13 00  ...x... .d4....
0030 00 00 13 00 00 00 04 00 00 00 01 00           .....
```

File: ping-gw.pcap 842 P: 8 D: 8 M: 0

# Tcpdump (console)

A terminal window titled 'vatn@dhcp-210: /home/vatn' with a menu bar containing 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal shows the execution of 'tcpdump -i eth1 -n', which captures 10 UDP packets from 192.168.0.1 to 239.255.255.250.1900. Summary statistics show 10 packets captured, 10 received by filter, and 0 dropped by kernel.

```
vatn@dhcp-210: /home/vatn
File Edit View Terminal Tabs Help
dhcp-210:/home/vatn# tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
21:33:42.695031 IP 192.168.0.1.4216 > 239.255.255.250.1900: UDP, length: 260
21:33:42.698551 IP 192.168.0.1.4216 > 239.255.255.250.1900: UDP, length: 278
21:33:42.702167 IP 192.168.0.1.4216 > 239.255.255.250.1900: UDP, length: 332
21:33:42.705660 IP 192.168.0.1.4217 > 239.255.255.250.1900: UDP, length: 324
21:33:42.708526 IP 192.168.0.1.4218 > 239.255.255.250.1900: UDP, length: 254
21:33:42.711849 IP 192.168.0.1.4218 > 239.255.255.250.1900: UDP, length: 296
21:33:42.715820 IP 192.168.0.1.4219 > 239.255.255.250.1900: UDP, length: 328
21:33:42.719335 IP 192.168.0.1.4220 > 239.255.255.250.1900: UDP, length: 274
21:33:42.723178 IP 192.168.0.1.4220 > 239.255.255.250.1900: UDP, length: 326
21:33:42.726841 IP 192.168.0.1.4221 > 239.255.255.250.1900: UDP, length: 320

10 packets captured
10 packets received by filter
0 packets dropped by kernel
dhcp-210:/home/vatn#
```



# Filtering out relevant traffic

## Capture filters:

- Smaller files
- “Tcpdump”-syntax,
- See ethereal “help” or “man tcpdump” for more info

[live capture]

## Display filters:

- Displays a subset of the captured data
- “Ethereal” has its own syntax, see ethereal “help”

[apply display filter]

# Observing DNS traffic

- Example: Testing if DNS look-up is faster the 2<sup>nd</sup> time. (My home network, WLAN router attached via ADSL to ISP)
- First test: looking up “[www.it.kth.se](http://www.it.kth.se)”
  - Files: [dns1.pcap, dns2.pcap]
  - Surprise, no caching effect was observed. Strange!
- Second test: looking up “[www.whitehouse.gov](http://www.whitehouse.gov)”
  - File: [dns-whitehouse.pcap]
  - The delay was even larger the 2<sup>nd</sup> time. For further investigation!!

# Example with DHCP traffic

- Dynamic Host Configuration protocol (DHCP, RFC 2131) enables a host to dynamically acquire an IP address as well as other relevant parameters from a DHCP server.
  - 4-way message exchange (Discover, Offer, Request, and Reply)
    - Should be fast, right?
    - Will we see additional messages?
- Files [dhcp-linux.pcap, dhcp-winxp.pcap]



# Analyzing measured data

Commonly one wants to analyze measured data, compute delays, calculate statics etc.

Many alternatives exist:

- Scripts (perl, awk)
- Spreadsheets
- Network analysis tools, e.g., “tcptrace”. See tcpdump and ethereal pages for more info.



Time-sequence plot created by tcptrace and displayed by xplot

# Eavesdropping attacks

- Do you send confidential information in clear-text?
- Network security is important, in particular when sending data over wireless links.

File: [web-mail.pcap]

