tcpdump:
network traffic capture

David Morgan

© David Morgan 2003-14

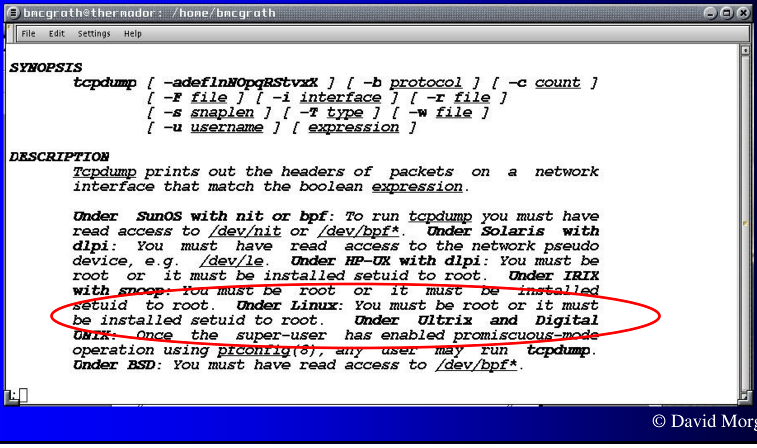
The Big Daddy of Open Source Capture

- tcpdump is the core Open Source packet sniffer program
- simple, text based program
- many other programs (such as Ethereal) that use the same file-save format can be used to display or interpret tcpdump files

© David Morgan 2003-14

Many options

Must be root to run



```
SYNOPSIS
       tcpdump [ -adeflnNOpqRStvxX ] [ -b protocol ] [ -c count ]
               [ -F file ] [ -i interface ] [ -r file ]
               [ -s snaplen ] [ -T type ] [ -w file ]
               [ -u username ] [ expression ]

DESCRIPTION
       Tcpdump prints out the headers of  packets  on  a  network
       interface that match the boolean expression.

       Under  SunOS with nit or bpf: To run tcpdump you must have
       read access to /dev/nit or /dev/bpf*.  Under Solaris  with
       dlpi:  You  must  have  read  access to the network pseudo
       device, e.g.  /dev/le.  Under HP-UX with dlpi: You must be
       root  or  it must be installed setuid to root.  Under IRIX
       with snoop: You must be  root  or  it  must  be  installed
       setuid  to root.  Under Linux: You must be root or it must
       be installed setuid to root.   Under  Ultrix  and  Digital
       UNIX:  Once  the  super-user  has enabled promiscuous-mode
       operation using pfconfig(8), any  user  may  run  tcpdump.
       Under BSD: You must have read access to /dev/bpf*.

:
```

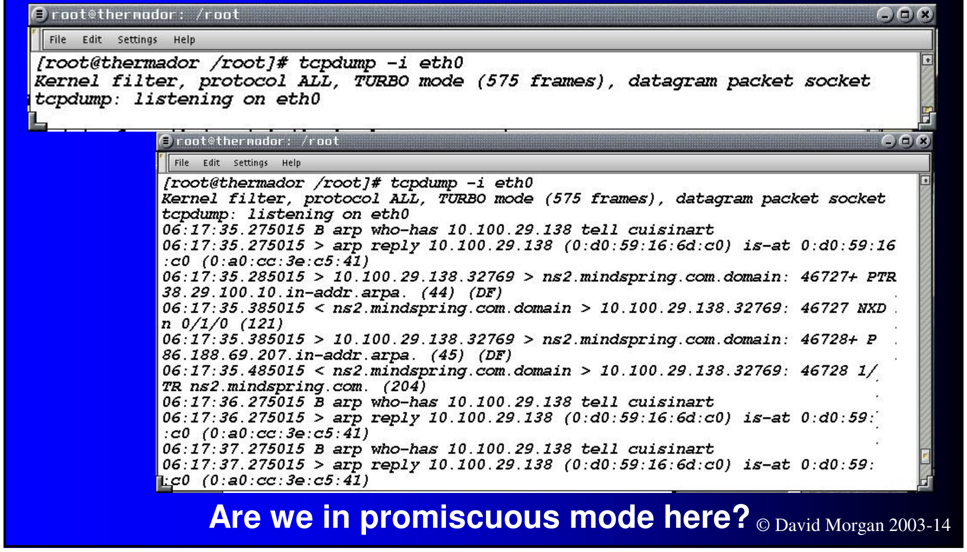
© David Morgan 2003-14

Some of them

Category	Option	Description
what to capture	-c	count of packets to capture
	-p	just mine (alternatively everyone’s)
where to capture	-i	interface specification
what to show	-t	omit timestamp
	-q	quiet – minimal output
	-v	verbose
	-vv	loquacious
	-vvv	blabby
	-x	packet content as well as header
how to show	-n	no address-to-name conversion
	-nn	nor port/protocol-to-name conversion
save/restore	-w	write capture to file
	-r	replay previous capture from file

© David Morgan 2003-14

tcpdump -i <interface>



```
[root@thermador /root]# tcpdump -i eth0
Kernel filter, protocol ALL, TURBO mode (575 frames), datagram packet socket
tcpdump: listening on eth0
```

```
[root@thermador /root]# tcpdump -i eth0
Kernel filter, protocol ALL, TURBO mode (575 frames), datagram packet socket
tcpdump: listening on eth0
06:17:35.275015 B arp who-has 10.100.29.138 tell cuisinart
06:17:35.275015 > arp reply 10.100.29.138 (0:d0:59:16:6d:c0) is-at 0:d0:59:16
:c0 (0:a0:cc:3e:c5:41)
06:17:35.285015 > 10.100.29.138.32769 > ns2.mindspring.com.domain: 46727+ PTR
38.29.100.10.in-addr.arpa. (44) (DF)
06:17:35.385015 < ns2.mindspring.com.domain > 10.100.29.138.32769: 46727 NXD
n 0/1/0 (121)
06:17:35.385015 > 10.100.29.138.32769 > ns2.mindspring.com.domain: 46728+ P
86.188.69.207.in-addr.arpa. (45) (DF)
06:17:35.485015 < ns2.mindspring.com.domain > 10.100.29.138.32769: 46728 1/
TR ns2.mindspring.com. (204)
06:17:36.275015 B arp who-has 10.100.29.138 tell cuisinart
06:17:36.275015 > arp reply 10.100.29.138 (0:d0:59:16:6d:c0) is-at 0:d0:59:
:c0 (0:a0:cc:3e:c5:41)
06:17:37.275015 B arp who-has 10.100.29.138 tell cuisinart
06:17:37.275015 > arp reply 10.100.29.138 (0:d0:59:16:6d:c0) is-at 0:d0:59:
:c0 (0:a0:cc:3e:c5:41)
```

Are we in promiscuous mode here?

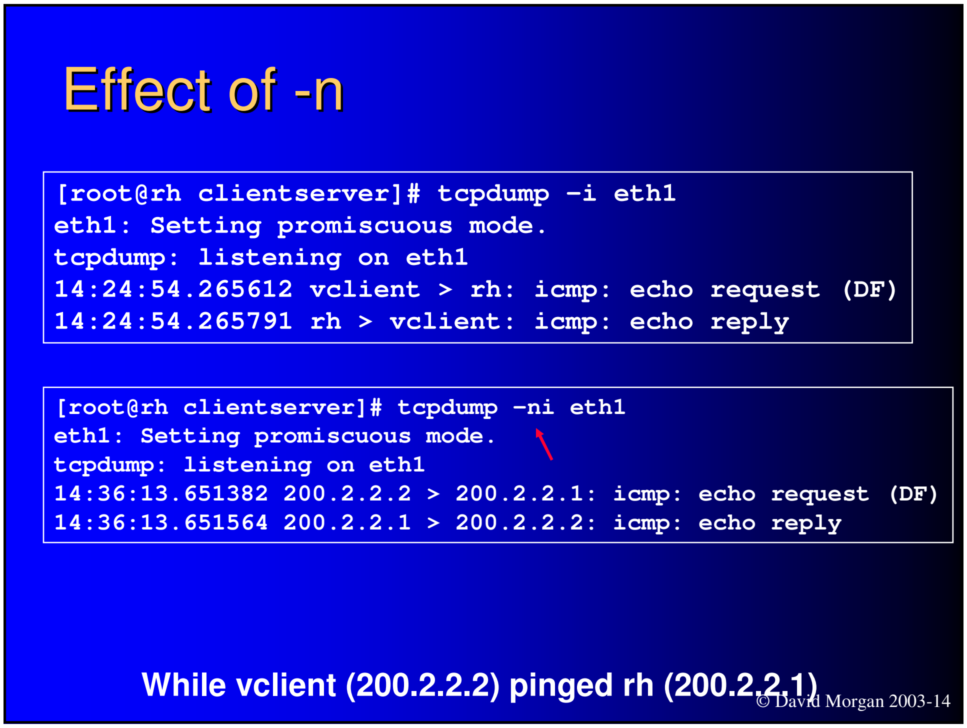
© David Morgan 2003-14

Capturing a ping

```
[root@rh clientserver]# tcpdump -i eth1
eth1: Setting promiscuous mode.
tcpdump: listening on eth1
14:24:54.265612 vclient > rh: icmp: echo request (DF)
14:24:54.265791 rh > vclient: icmp: echo reply
```

While vclient pinged rh

© David Morgan 2003-14



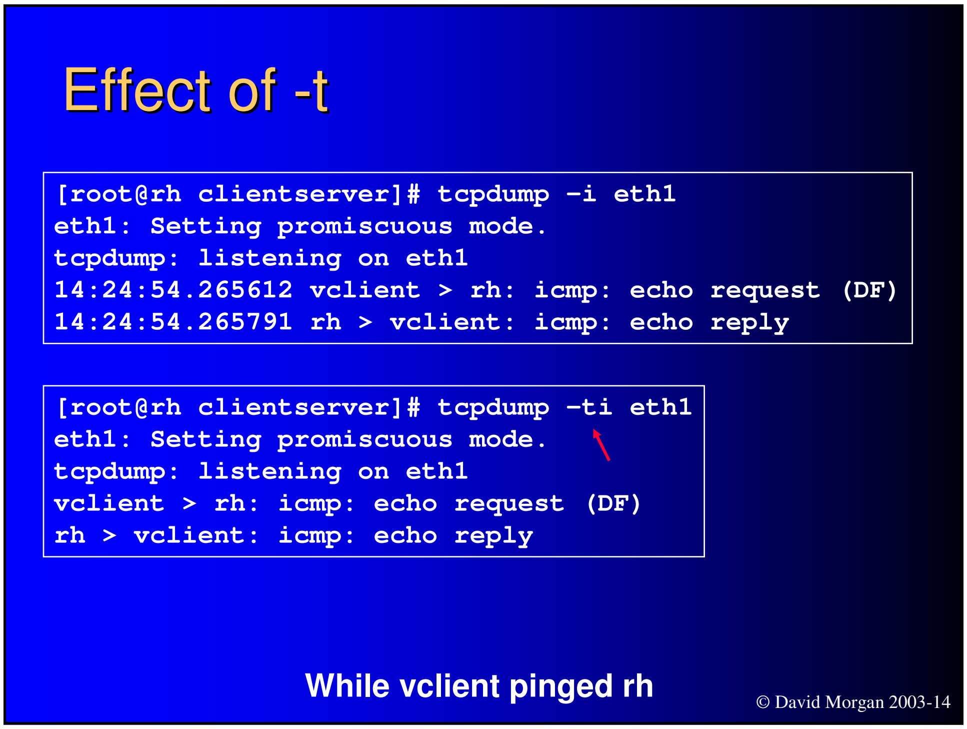
Effect of -n

```
[root@rh clientserver]# tcpdump -i eth1
eth1: Setting promiscuous mode.
tcpdump: listening on eth1
14:24:54.265612 vclient > rh: icmp: echo request (DF)
14:24:54.265791 rh > vclient: icmp: echo reply
```

```
[root@rh clientserver]# tcpdump -ni eth1
eth1: Setting promiscuous mode.
tcpdump: listening on eth1
14:36:13.651382 200.2.2.2 > 200.2.2.1: icmp: echo request (DF)
14:36:13.651564 200.2.2.1 > 200.2.2.2: icmp: echo reply
```

While vclient (200.2.2.2) pinged rh (200.2.2.1)

© David Morgan 2003-14



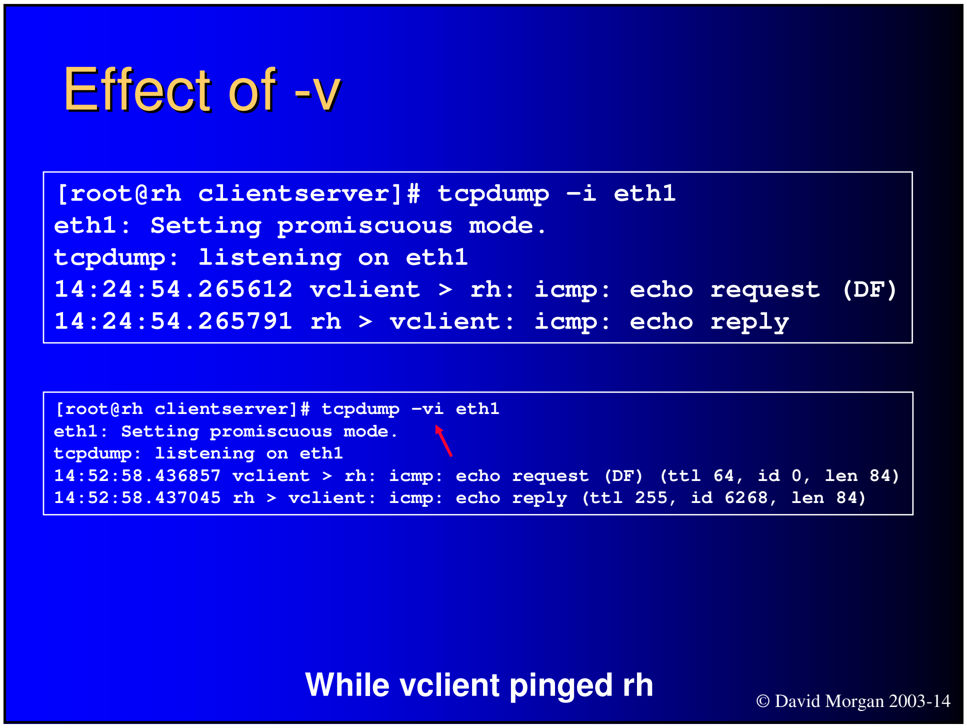
Effect of -t

```
[root@rh clientserver]# tcpdump -i eth1
eth1: Setting promiscuous mode.
tcpdump: listening on eth1
14:24:54.265612 vclient > rh: icmp: echo request (DF)
14:24:54.265791 rh > vclient: icmp: echo reply
```

```
[root@rh clientserver]# tcpdump -ti eth1
eth1: Setting promiscuous mode.
tcpdump: listening on eth1
vclient > rh: icmp: echo request (DF)
rh > vclient: icmp: echo reply
```

While vclient pinged rh

© David Morgan 2003-14



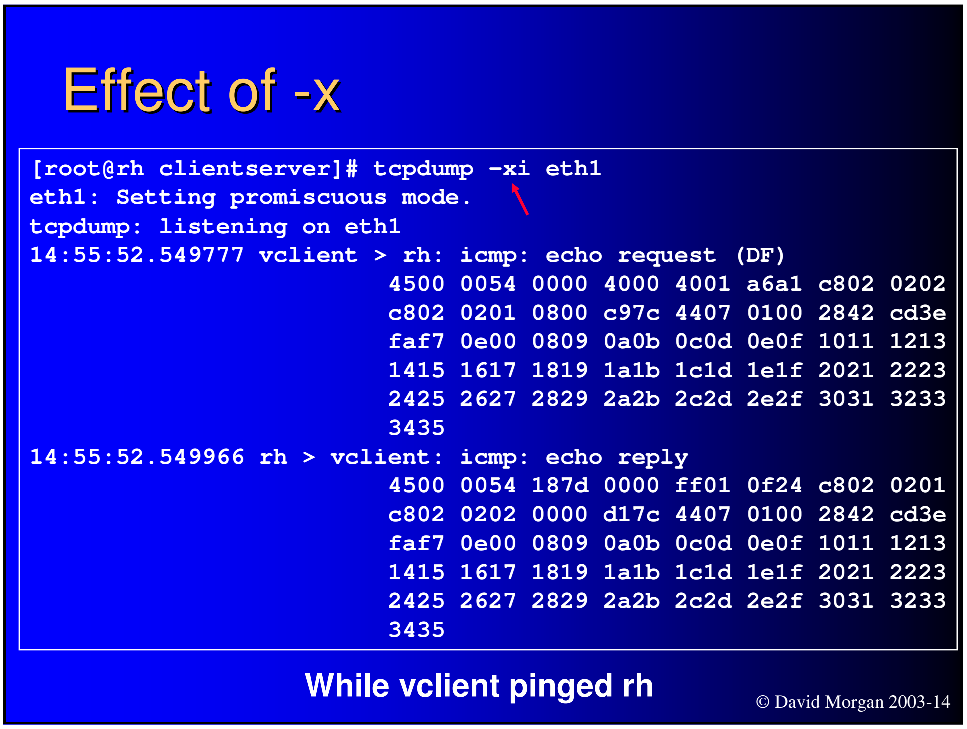
Effect of -v

```
[root@rh clientserver]# tcpdump –i eth1
eth1: Setting promiscuous mode.
tcpdump: listening on eth1
14:24:54.265612 vclient > rh: icmp: echo request (DF)
14:24:54.265791 rh > vclient: icmp: echo reply
```

```
[root@rh clientserver]# tcpdump –vi eth1
eth1: Setting promiscuous mode.
tcpdump: listening on eth1
14:52:58.436857 vclient > rh: icmp: echo request (DF) (ttl 64, id 0, len 84)
14:52:58.437045 rh > vclient: icmp: echo reply (ttl 255, id 6268, len 84)
```

While vclient pinged rh

© David Morgan 2003-14

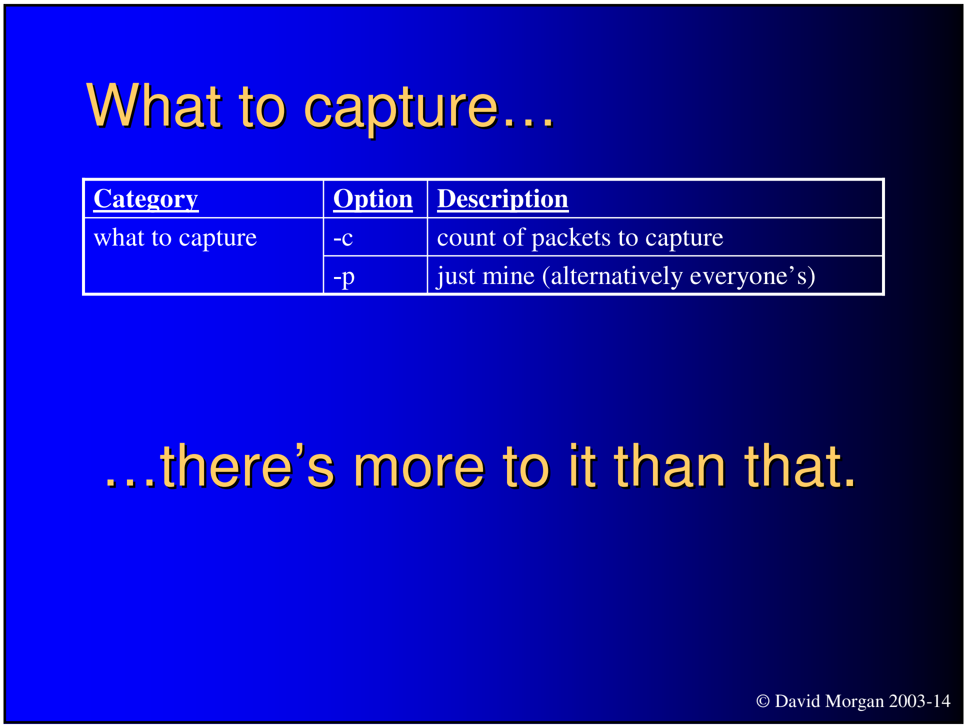


Effect of -x

```
[root@rh clientserver]# tcpdump –xi eth1
eth1: Setting promiscuous mode.
tcpdump: listening on eth1
14:55:52.549777 vclient > rh: icmp: echo request (DF)
                         4500 0054 0000 4000 4001 a6a1 c802 0202
                         c802 0201 0800 c97c 4407 0100 2842 cd3e
                         faf7 0e00 0809 0a0b 0c0d 0e0f 1011 1213
                         1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
                         2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
                         3435
14:55:52.549966 rh > vclient: icmp: echo reply
                         4500 0054 187d 0000 ff01 0f24 c802 0201
                         c802 0202 0000 d17c 4407 0100 2842 cd3e
                         faf7 0e00 0809 0a0b 0c0d 0e0f 1011 1213
                         1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
                         2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
                         3435
```

While vclient pinged rh

© David Morgan 2003-14

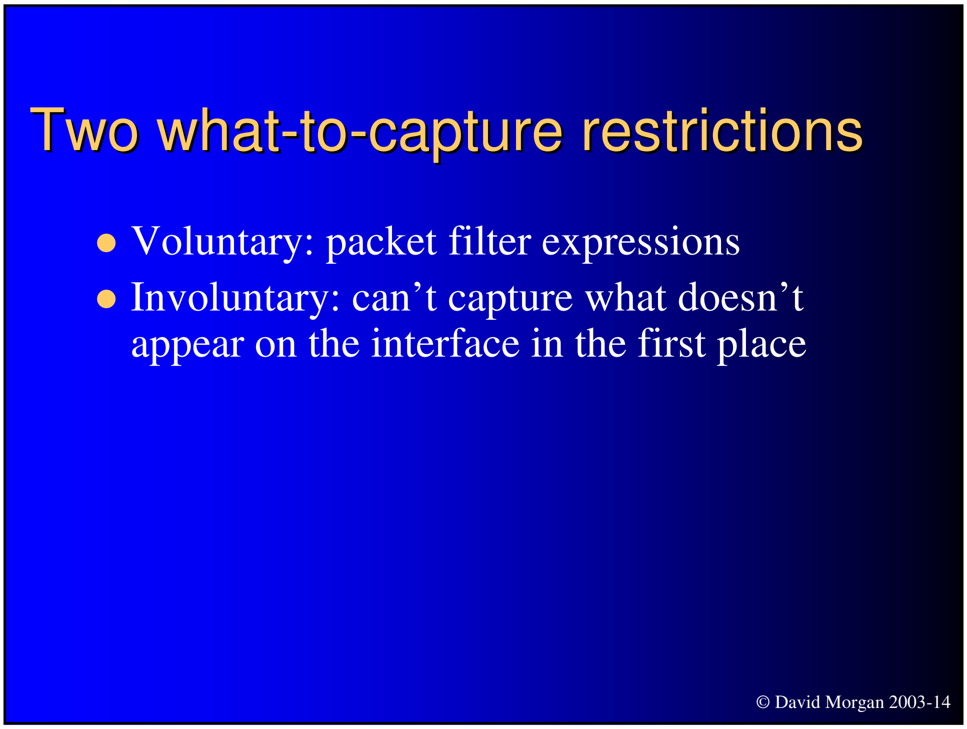


What to capture…

Category	Option	Description
what to capture	-c	count of packets to capture
	-p	just mine (alternatively everyone’s)

…there’s more to it than that.

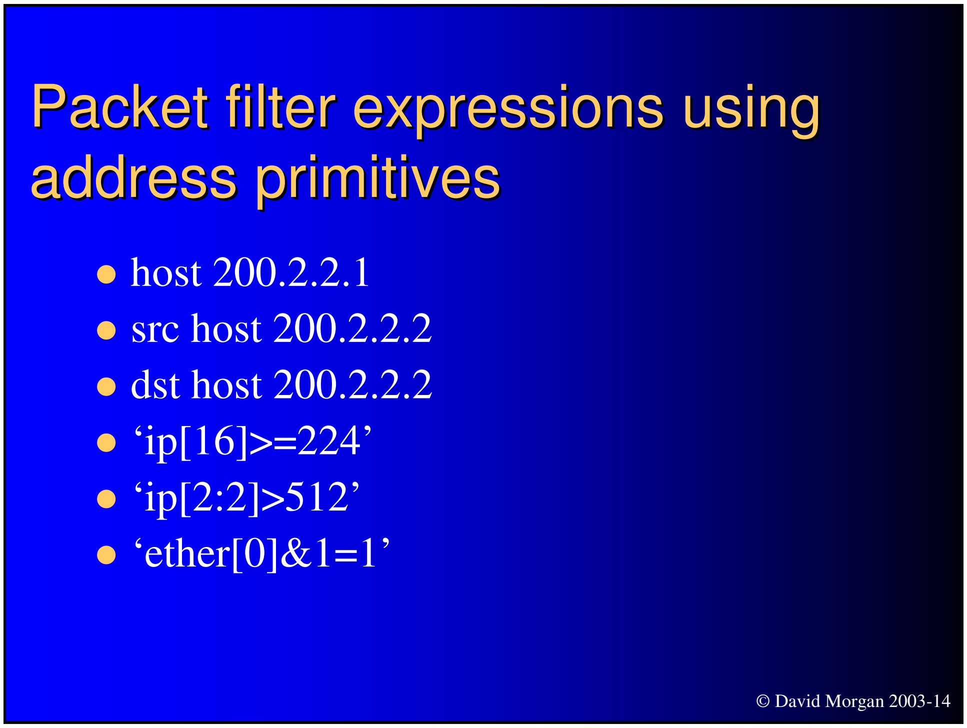
© David Morgan 2003-14



Two what-to-capture restrictions

- Voluntary: packet filter expressions
- Involuntary: can’t capture what doesn’t appear on the interface in the first place

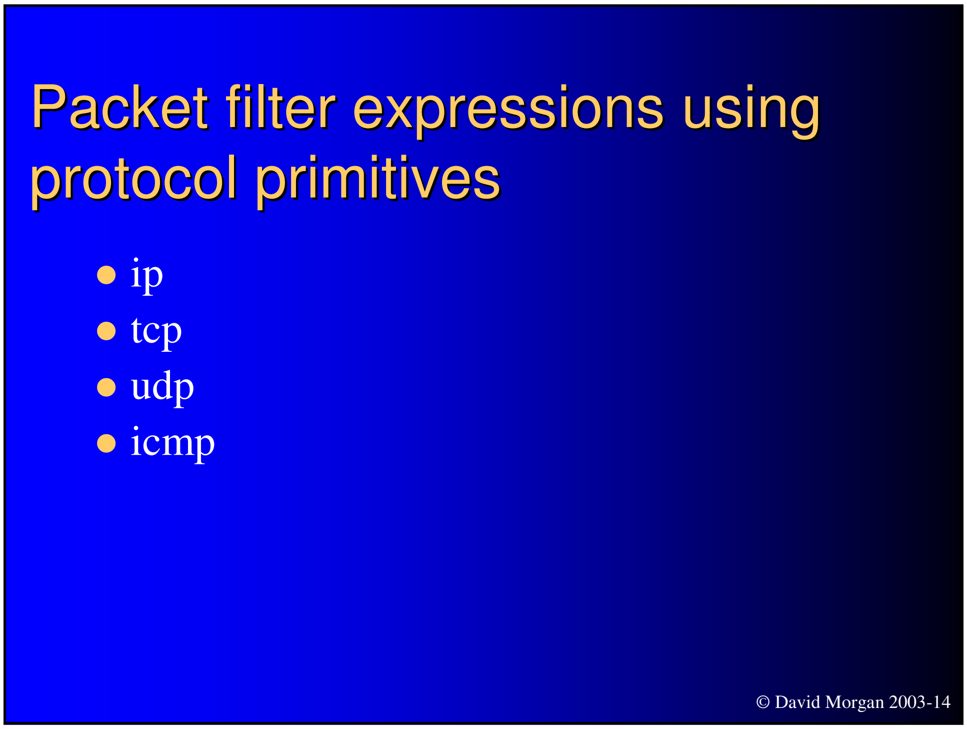
© David Morgan 2003-14



Packet filter expressions using address primitives

- host 200.2.2.1
- src host 200.2.2.2
- dst host 200.2.2.2
- ‘ip[16]>=224’
- ‘ip[2:2]>512’
- ‘ether[0]&1=1’

© David Morgan 2003-14



Packet filter expressions using protocol primitives

- ip
- tcp
- udp
- icmp

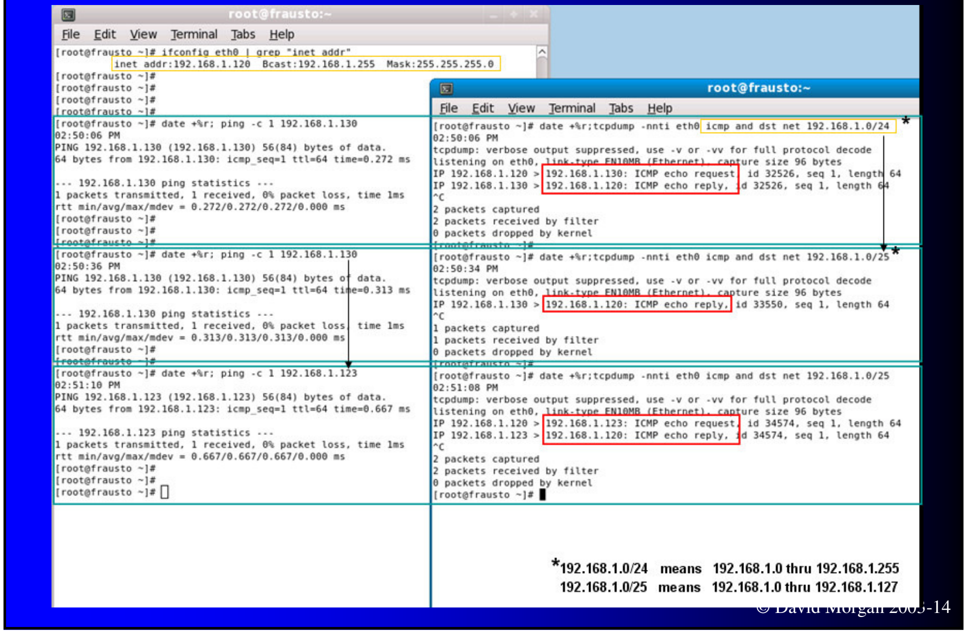
© David Morgan 2003-14

Booleans

- and
- or
- not

© David Morgan 2003-14

Filter example



```
[root@frausto ~]# ifconfig eth0 | grep "inet addr"
          inet addr:192.168.1.120  Bcast:192.168.1.255  Mask:255.255.255.0
[root@frausto ~]#
[root@frausto ~]#
[root@frausto ~]#
[root@frausto ~]#
[root@frausto ~]# date +%r; ping -c 1 192.168.1.130
02:50:06 PM
PING 192.168.1.130 (192.168.1.130) 56(84) bytes of data.
64 bytes from 192.168.1.130: icmp_seq=1 ttl=64 time=0.272 ms

--- 192.168.1.130 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 1ms
rtt min/avg/max/mdev = 0.272/0.272/0.272/0.000 ms
[root@frausto ~]#
[root@frausto ~]#
[root@frausto ~]#
[root@frausto ~]# date +%r; ping -c 1 192.168.1.130
02:50:36 PM
PING 192.168.1.130 (192.168.1.130) 56(84) bytes of data.
64 bytes from 192.168.1.130: icmp_seq=1 ttl=64 time=0.313 ms

--- 192.168.1.130 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 1ms
rtt min/avg/max/mdev = 0.313/0.313/0.313/0.000 ms
[root@frausto ~]#
[root@frausto ~]#
[root@frausto ~]# date +%r; ping -c 1 192.168.1.123
02:51:10 PM
PING 192.168.1.123 (192.168.1.123) 56(84) bytes of data.
64 bytes from 192.168.1.123: icmp_seq=1 ttl=64 time=0.667 ms

--- 192.168.1.123 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 1ms
rtt min/avg/max/mdev = 0.667/0.667/0.667/0.000 ms
[root@frausto ~]#
[root@frausto ~]#
[root@frausto ~]#
```

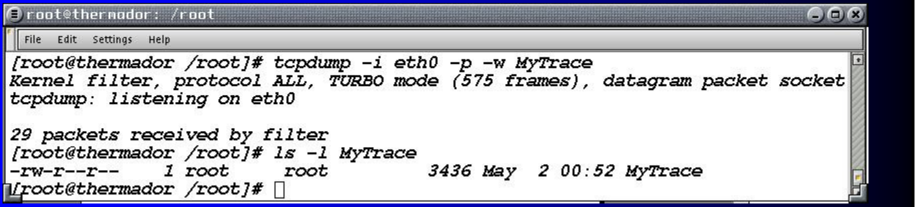
```
[root@frausto ~]# date +%r;tcpdump -nnti eth0 icmp and dst net 192.168.1.0/24 *
02:50:06 PM
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
IP 192.168.1.120 > 192.168.1.130: ICMP echo request, id 32526, seq 1, length 64
IP 192.168.1.130 > 192.168.1.120: ICMP echo reply, id 32526, seq 1, length 64
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
[root@frausto ~]#
[root@frausto ~]# date +%r;tcpdump -nnti eth0 icmp and dst net 192.168.1.0/25 *
02:50:34 PM
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
IP 192.168.1.130 > 192.168.1.120: ICMP echo reply, id 33550, seq 1, length 64
^C
1 packets captured
1 packets received by filter
0 packets dropped by kernel
[root@frausto ~]#
[root@frausto ~]# date +%r;tcpdump -nnti eth0 icmp and dst net 192.168.1.0/25
02:51:08 PM
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
IP 192.168.1.120 > 192.168.1.123: ICMP echo request, id 34574, seq 1, length 64
IP 192.168.1.123 > 192.168.1.120: ICMP echo reply, id 34574, seq 1, length 64
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
[root@frausto ~]#
```

*192.168.1.0/24 means 192.168.1.0 thru 192.168.1.255
192.168.1.0/25 means 192.168.1.0 thru 192.168.1.127

© David Morgan 2003-14

Write to File

-w <file name> will redirect output to a file



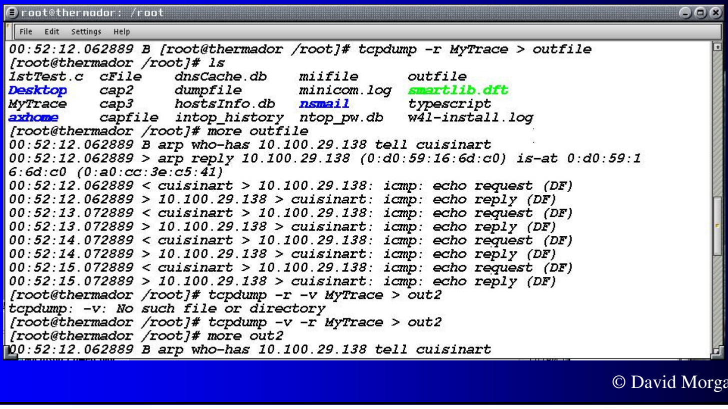
```
[root@thermador /root]# tcpdump -i eth0 -p -w MyTrace
Kernel filter, protocol ALL, TURBO mode (575 frames), datagram packet socket
tcpdump: listening on eth0

29 packets received by filter
[root@thermador /root]# ls -l MyTrace
-rw-r--r--    1 root     root         3436 May  2 00:52 MyTrace
[root@thermador /root]# 
```

© David Morgan 2003-14

Replay with -r option

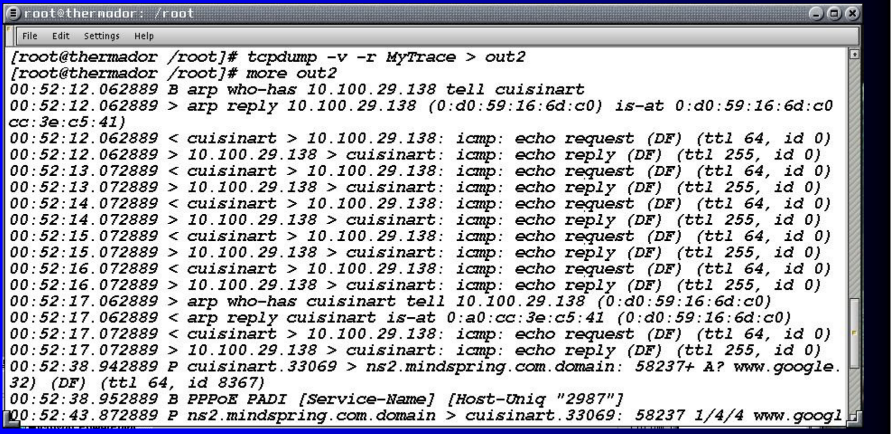
tcpdump -r <rawfile> <text file>



```
00:52:12.062889 B [root@thermador /root]# tcpdump -r MyTrace > outfile
[root@thermador /root]# ls
1stTest.c   cFile     dnsCache.db    miifile      outfile
Desktop     cap2      dumpfile       minicom.log  smartlib.dft
MyTrace     cap3      hostsInfo.db   nsmail       typescript
axhome      capfile   intop_history  ntop_pw.db   w41-install.log
[root@thermador /root]# more outfile
00:52:12.062889 B arp who-has 10.100.29.138 tell cuisinart
00:52:12.062889 > arp reply 10.100.29.138 (0:d0:59:16:6d:c0) is-at 0:d0:59:1
6:6d:c0 (0:a0:cc:3e:c5:41)
00:52:12.062889 < cuisinart > 10.100.29.138: icmp: echo request (DF)
00:52:12.062889 > 10.100.29.138 > cuisinart: icmp: echo reply (DF)
00:52:13.072889 < cuisinart > 10.100.29.138: icmp: echo request (DF)
00:52:13.072889 > 10.100.29.138 > cuisinart: icmp: echo reply (DF)
00:52:14.072889 < cuisinart > 10.100.29.138: icmp: echo request (DF)
00:52:14.072889 > 10.100.29.138 > cuisinart: icmp: echo reply (DF)
00:52:15.072889 < cuisinart > 10.100.29.138: icmp: echo request (DF)
00:52:15.072889 > 10.100.29.138 > cuisinart: icmp: echo reply (DF)
[root@thermador /root]# tcpdump -r -v MyTrace > out2
tcpdump: -v: No such file or directory
[root@thermador /root]# tcpdump -v -r MyTrace > out2
[root@thermador /root]# more out2
00:52:12.062889 B arp who-has 10.100.29.138 tell cuisinart
```

© David Morgan 2003-14

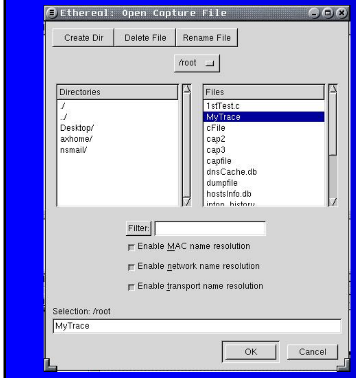
-v works on playback too



```
[root@thermador /root]# tcpdump -v -r MyTrace > out2
[root@thermador /root]# more out2
00:52:12.062889 B arp who-has 10.100.29.138 tell cuisinart
00:52:12.062889 > arp reply 10.100.29.138 (0:d0:59:16:6d:c0) is-at 0:d0:59:16:6d:c0
cc:3e:c5:41)
00:52:12.062889 < cuisinart > 10.100.29.138: icmp: echo request (DF) (ttl 64, id 0)
00:52:12.062889 > 10.100.29.138 > cuisinart: icmp: echo reply (DF) (ttl 255, id 0)
00:52:13.072889 < cuisinart > 10.100.29.138: icmp: echo request (DF) (ttl 64, id 0)
00:52:13.072889 > 10.100.29.138 > cuisinart: icmp: echo reply (DF) (ttl 255, id 0)
00:52:14.072889 < cuisinart > 10.100.29.138: icmp: echo request (DF) (ttl 64, id 0)
00:52:14.072889 > 10.100.29.138 > cuisinart: icmp: echo reply (DF) (ttl 255, id 0)
00:52:15.072889 < cuisinart > 10.100.29.138: icmp: echo request (DF) (ttl 64, id 0)
00:52:15.072889 > 10.100.29.138 > cuisinart: icmp: echo reply (DF) (ttl 255, id 0)
00:52:16.072889 < cuisinart > 10.100.29.138: icmp: echo request (DF) (ttl 64, id 0)
00:52:16.072889 > 10.100.29.138 > cuisinart: icmp: echo reply (DF) (ttl 255, id 0)
00:52:17.062889 > arp who-has cuisinart tell 10.100.29.138 (0:d0:59:16:6d:c0)
00:52:17.062889 < arp reply cuisinart is-at 0:a0:cc:3e:c5:41 (0:d0:59:16:6d:c0)
00:52:17.072889 < cuisinart > 10.100.29.138: icmp: echo request (DF) (ttl 64, id 0)
00:52:17.072889 > 10.100.29.138 > cuisinart: icmp: echo reply (DF) (ttl 255, id 0)
00:52:38.942889 P cuisinart.33069 > ns2.mindspring.com.domain: 58237+ A? www.google.
32) (DF) (ttl 64, id 8367)
00:52:38.952889 B PPPoE PADI [Service-Name] [Host-Uniq "2987"]
00:52:43.872889 P ns2.mindspring.com.domain > cuisinart.33069: 58237 1/4/4 www.googl
```

© David Morgan 2003-14

View file with Wireshark too



Directories: ./ ../ Desktop/ axhome/ nsmail/

Files: 1stTest.c, MyTrace, cFile, cap2, cap3, capfile, dnsCache.db, dumpfile, hostsInfo.db, intop_history

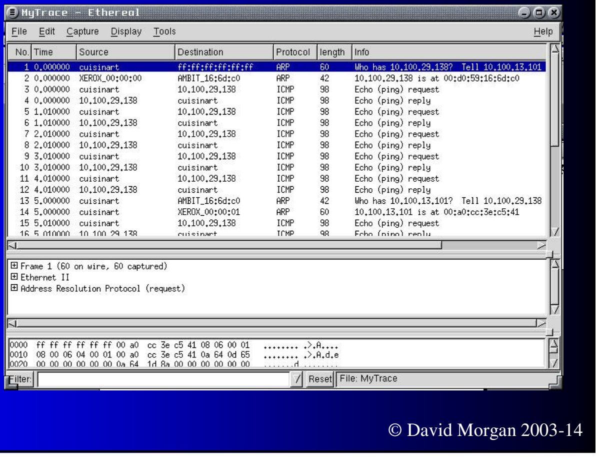
□ Enable MAC name resolution

□ Enable network name resolution

□ Enable transport name resolution

Selection: /root

MyTrace



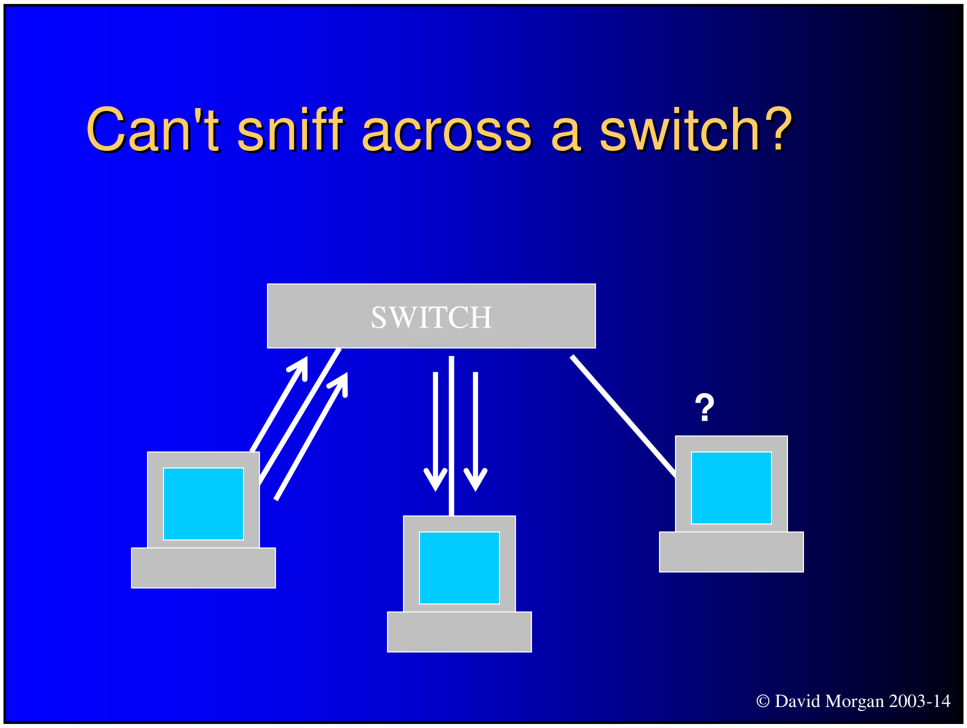
```
No. Time      Source          Destination        Protocol length Info
  1 0.000000  cuisinart       ff:ff:ff:ff:ff:ff  ARP      60     Who has 10.100.29.138?  Tell 10.100.13.101
  2 0.000000  XEROX_00:00:00  AMBIT_16:6d:c0     ARP      42     10.100.29.138 is at 00:d0:59:16:6d:c0
  3 0.000000  cuisinart       10.100.29.138      ICMP     98     Echo (ping) request
  4 0.000000  10.100.29.138   cuisinart          ICMP     98     Echo (ping) reply
  5 1.010000  cuisinart       10.100.29.138      ICMP     98     Echo (ping) request
  6 1.010000  10.100.29.138   cuisinart          ICMP     98     Echo (ping) reply
  7 2.010000  cuisinart       10.100.29.138      ICMP     98     Echo (ping) request
  8 2.010000  10.100.29.138   cuisinart          ICMP     98     Echo (ping) reply
  9 3.010000  cuisinart       10.100.29.138      ICMP     98     Echo (ping) request
 10 3.010000  10.100.29.138   cuisinart          ICMP     98     Echo (ping) reply
 11 4.010000  cuisinart       10.100.29.138      ICMP     98     Echo (ping) request
 12 4.010000  10.100.29.138   cuisinart          ICMP     98     Echo (ping) reply
 13 5.000000  cuisinart       AMBIT_16:6d:c0     ARP      42     Who has 10.100.13.101?  Tell 10.100.29.138
 14 5.000000  cuisinart       XEROX_00:00:01     ARP      60     10.100.13.101 is at 00:a0:cc:3e:c5:41
 15 5.010000  cuisinart       10.100.29.138      ICMP     98     Echo (ping) request
 16 5.010000  10.100.29.138   cuisinart          ICMP     98     Echo (ping) reply

⊞ Frame 1 (60 on wire, 60 captured)
⊞ Ethernet II
⊞ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 00 a0  cc 3e c5 41 08 06 00 01   ........ .>.A....
0010  08 00 06 04 00 01 00 a0  cc 3e c5 41 0a 64 0d 65   ........ .>.A.d.e
0020  00 00 00 00 00 00 0a 64  1d 8a 00 00 00 00 00 00   .......d ........
```

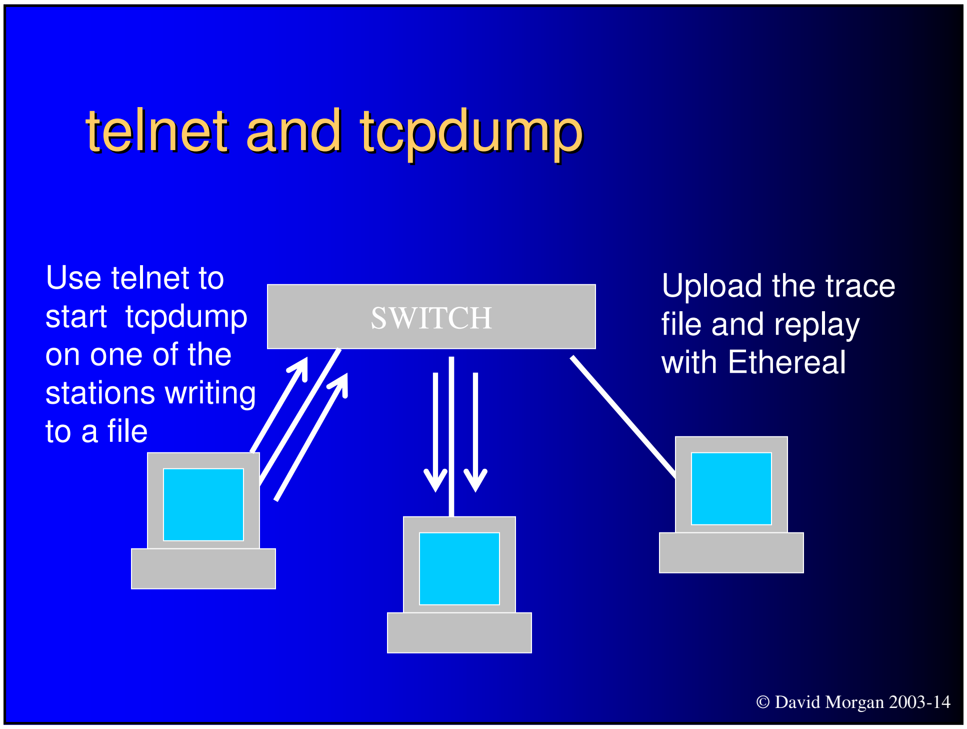
File: MyTrace

© David Morgan 2003-14



Can't sniff across a switch?

© David Morgan 2003-14



telnet and tcpdump

Use telnet to start tcpdump on one of the stations writing to a file

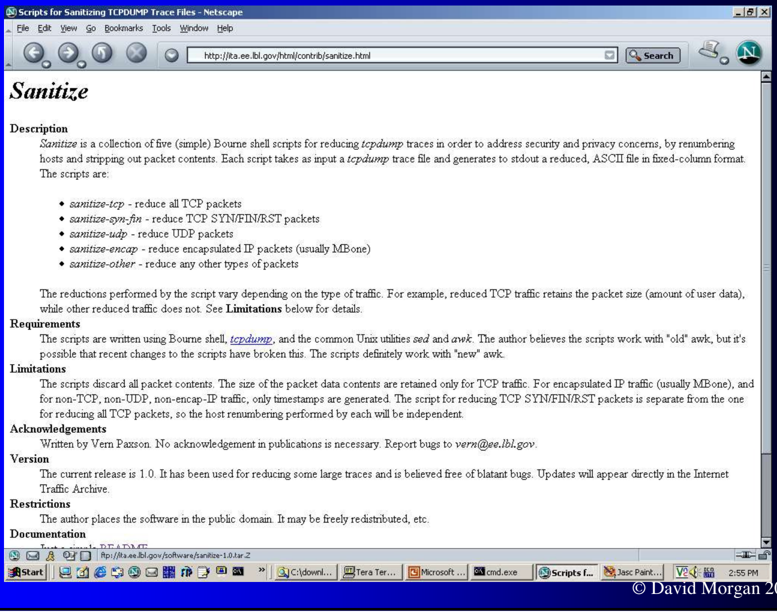
Upload the trace file and replay with Ethereal

© David Morgan 2003-14

Analysis tools for dump files

- sanitize
- tcpdpriv
- tcpflow
- tcp-reduce
- tcpshow
- tcpslice
- trafshow

sanitize


Scripts for Sanitizing TCPDUMP Trace Files - Netscape
File Edit View Go Bookmarks Tools Window Help
http://ita.ee.lbl.gov/html/contrib/sanitize.html
Search
ftp://ita.ee.lbl.gov/software/sanitize-1.0.tar.Z
Start
C:\downl...
Tera Ter...
Microsoft ...
cmd.exe
Scripts f...
Jasc Paint...
2:55 PM

Sanitize

Description

Sanitize is a collection of five (simple) Bourne shell scripts for reducing *tcpdump* traces in order to address security and privacy concerns, by renumbering hosts and stripping out packet contents. Each script takes as input a *tcpdump* trace file and generates to stdout a reduced, ASCII file in fixed-column format. The scripts are:

- *sanitize-tcp* - reduce all TCP packets
- *sanitize-syn-fin* - reduce TCP SYN/FIN/RST packets
- *sanitize-udp* - reduce UDP packets
- *sanitize-encap* - reduce encapsulated IP packets (usually MBone)
- *sanitize-other* - reduce any other types of packets

The reductions performed by the script vary depending on the type of traffic. For example, reduced TCP traffic retains the packet size (amount of user data), while other reduced traffic does not. See **Limitations** below for details.

Requirements

The scripts are written using Bourne shell, *tcpdump*, and the common Unix utilities *sed* and *awk*. The author believes the scripts work with "old" awk, but it's possible that recent changes to the scripts have broken this. The scripts definitely work with "new" awk.

Limitations

The scripts discard all packet contents. The size of the packet data contents are retained only for TCP traffic. For encapsulated IP traffic (usually MBone), and for non-TCP, non-UDP, non-encap-IP traffic, only timestamps are generated. The script for reducing TCP SYN/FIN/RST packets is separate from the one for reducing all TCP packets, so the host renumbering performed by each will be independent.

Acknowledgements

Written by Vern Paxson. No acknowledgement in publications is necessary. Report bugs to *vern@ee.lbl.gov*.

Version

The current release is 1.0. It has been used for reducing some large traces and is believed free of blatant bugs. Updates will appear directly in the Internet Traffic Archive.

Restrictions

The author places the software in the public domain. It may be freely redistributed, etc.

Documentation

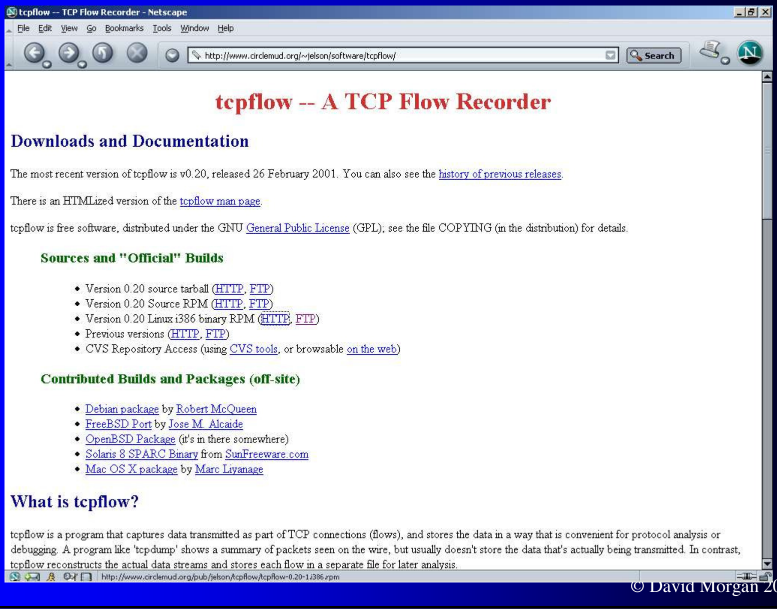
Just a simple README

sanitize

- Collection of shell scripts
  - sanitize-tcp
  - sanitize-syn-fin
  - sanitize-udp
  - sanitize-encap
  - sanitize-other
- Each filters out all packets except…
- Rewrites remaining packets
  - less info
  - renumbered (not actual) addresses

© David Morgan 2003-14

tcpflow


tcpflow -- TCP Flow Recorder - Netscape
File Edit View Go Bookmarks Tools Window Help
http://www.circlemud.org/~jelson/software/tcpflow/
Search
http://www.circlemud.org/pub/jelson/tcpflow/tcpflow-0.20-1.i386.rpm

tcpflow -- A TCP Flow Recorder

Downloads and Documentation

The most recent version of tcpflow is v0.20, released 26 February 2001. You can also see the history of previous releases.

There is an HTMLized version of the tcpflow man page.

tcpflow is free software, distributed under the GNU General Public License (GPL); see the file COPYING (in the distribution) for details.

Sources and "Official" Builds

- Version 0.20 source tarball (HTTP, FTP)
- Version 0.20 Source RPM (HTTP, FTP)
- Version 0.20 Linux i386 binary RPM (HTTP, FTP)
- Previous versions (HTTP, FTP)
- CVS Repository Access (using CVS tools, or browsable on the web)

Contributed Builds and Packages (off-site)

- Debian package by Robert McQueen
- FreeBSD Port by Jose M. Alcaide
- OpenBSD Package (it's in there somewhere)
- Solaris 8 SPARC Binary from SunFreeware.com
- Mac OS X package by Marc Liyanage

What is tcpflow?

tcpflow is a program that captures data transmitted as part of TCP connections (flows), and stores the data in a way that is convenient for protocol analysis or debugging. A program like 'tcpdump' shows a summary of packets seen on the wire, but usually doesn't store the data that's actually being transmitted. In contrast, tcpflow reconstructs the actual data streams and stores each flow in a separate file for later analysis.

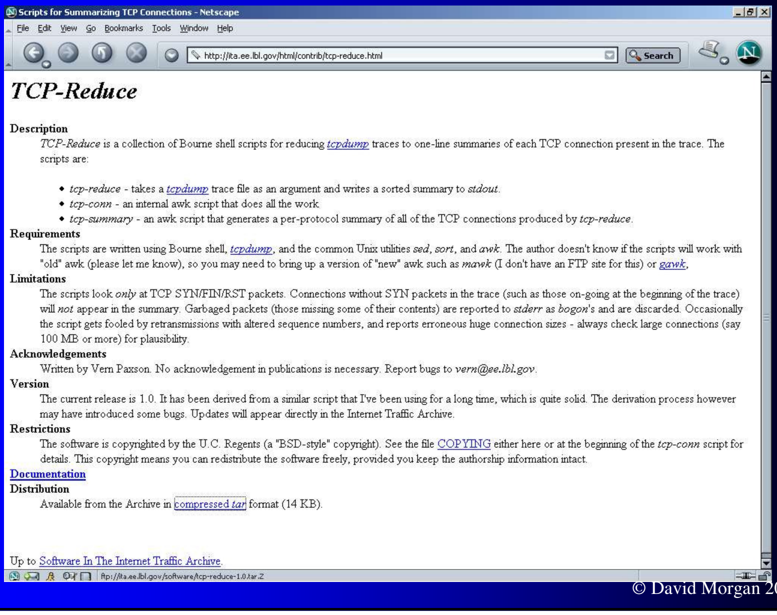
© David Morgan 2003-14

tcpflow

- apply to tcpdump-style capture file
- segregates traffic by TCP connection
  - – uniquely identified by quartet of 2 IP addresses and 2 ports
- extracts data only, from each connection
- stores it in separate files whose names reflect the connection

© David Morgan 2003-14

tcp-reduce


Scripts for Summarizing TCP Connections - Netscape
File Edit View Go Bookmarks Tools Window Help
http://ita.ee.lbl.gov/html/contrib/tcp-reduce.html
Search
ftp://ita.ee.lbl.gov/software/tcp-reduce-1.0.tar.Z

TCP-Reduce

Description

TCP-Reduce is a collection of Bourne shell scripts for reducing *tcpdump* traces to one-line summaries of each TCP connection present in the trace. The scripts are:

- *tcp-reduce* - takes a *tcpdump* trace file as an argument and writes a sorted summary to *stdout*.
- *tcp-conn* - an internal awk script that does all the work
- *tcp-summary* - an awk script that generates a per-protocol summary of all of the TCP connections produced by *tcp-reduce*.

Requirements

The scripts are written using Bourne shell, *tcpdump*, and the common Unix utilities *sed*, *sort*, and *awk*. The author doesn't know if the scripts will work with "old" awk (please let me know), so you may need to bring up a version of "new" awk such as *mawk* (I don't have an FTP site for this) or *gawk*,

Limitations

The scripts look *only* at TCP SYN/FIN/RST packets. Connections without SYN packets in the trace (such as those on-going at the beginning of the trace) will *not* appear in the summary. Garbaged packets (those missing some of their contents) are reported to *stderr* as *bogon*'s and are discarded. Occasionally the script gets fooled by retransmissions with altered sequence numbers, and reports erroneous huge connection sizes - always check large connections (say 100 MB or more) for plausibility.

Acknowledgements

Written by Vern Paxson. No acknowledgement in publications is necessary. Report bugs to *vern@ee.lbl.gov*.

Version

The current release is 1.0. It has been derived from a similar script that I've been using for a long time, which is quite solid. The derivation process however may have introduced some bugs. Updates will appear directly in the Internet Traffic Archive.

Restrictions

The software is copyrighted by the U.C. Regents (a "BSD-style" copyright). See the file COPYING either here or at the beginning of the *tcp-conn* script for details. This copyright means you can redistribute the software freely, provided you keep the authorship information intact.

Documentation

Distribution

Available from the Archive in compressed *tar* format (14 KB).

Up to Software In The Internet Traffic Archive.

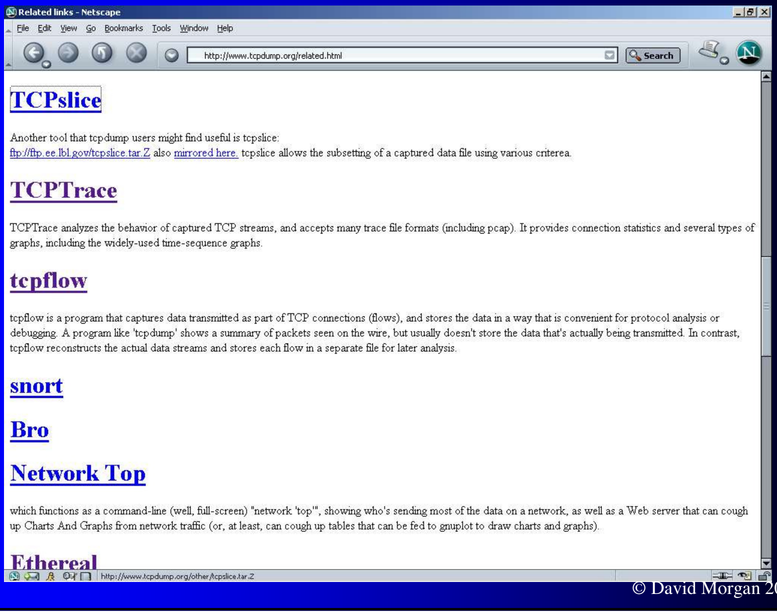
© David Morgan 2003-14

tcp-reduce

- single-line summary, each TCP connection
- information fields
  - time and duration
  - protocol
  - bytes sent, each side
  - TCP state at termination

© David Morgan 2003-14

tcpslice


Related links - Netscape
File Edit View Go Bookmarks Tools Window Help
http://www.tcpdump.org/related.html
Search
http://www.tcpdump.org/other/tcpslice.tar.Z

TCPslice

Another tool that tcpdump users might find useful is tcpslice:
ftp://ftp.ee.lbl.gov/tcpslice.tar.Z also mirrored here. tcpslice allows the subsetting of a captured data file using various criterea.

TCPTrace

TCPTrace analyzes the behavior of captured TCP streams, and accepts many trace file formats (including pcap). It provides connection statistics and several types of graphs, including the widely-used time-sequence graphs.

tcpflow

tcpflow is a program that captures data transmitted as part of TCP connections (flows), and stores the data in a way that is convenient for protocol analysis or debugging. A program like 'tcpdump' shows a summary of packets seen on the wire, but usually doesn't store the data that's actually being transmitted. In contrast, tcpflow reconstructs the actual data streams and stores each flow in a separate file for later analysis.

snort

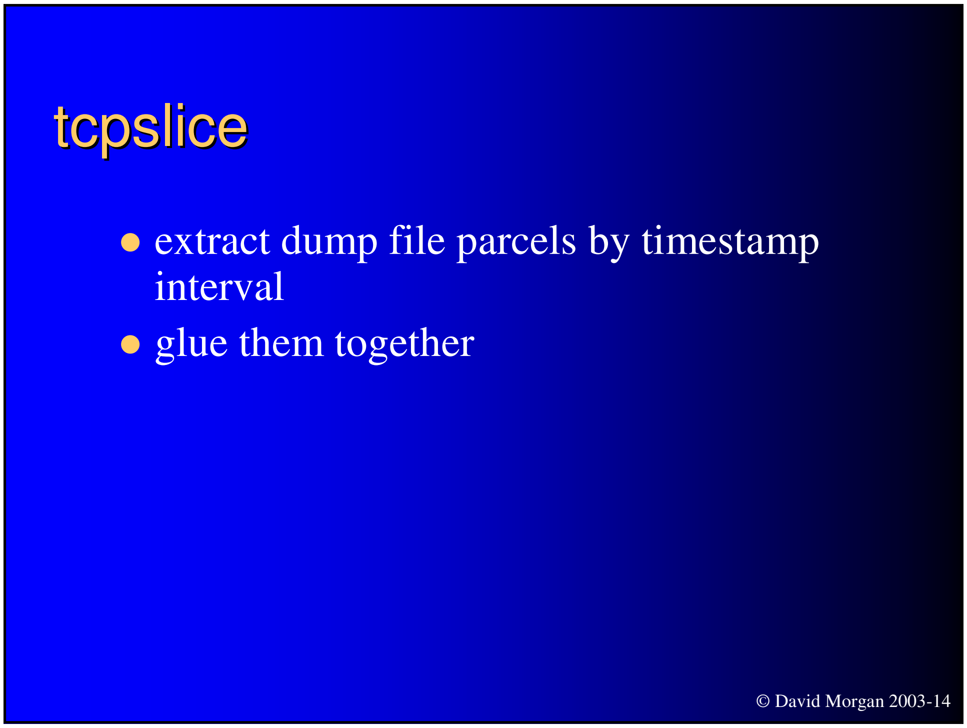
Bro

Network Top

which functions as a command-line (well, full-screen) "network 'top'", showing who's sending most of the data on a network, as well as a Web server that can cough up Charts And Graphs from network traffic (or, at least, can cough up tables that can be fed to gnuplot to draw charts and graphs).

Ethereal

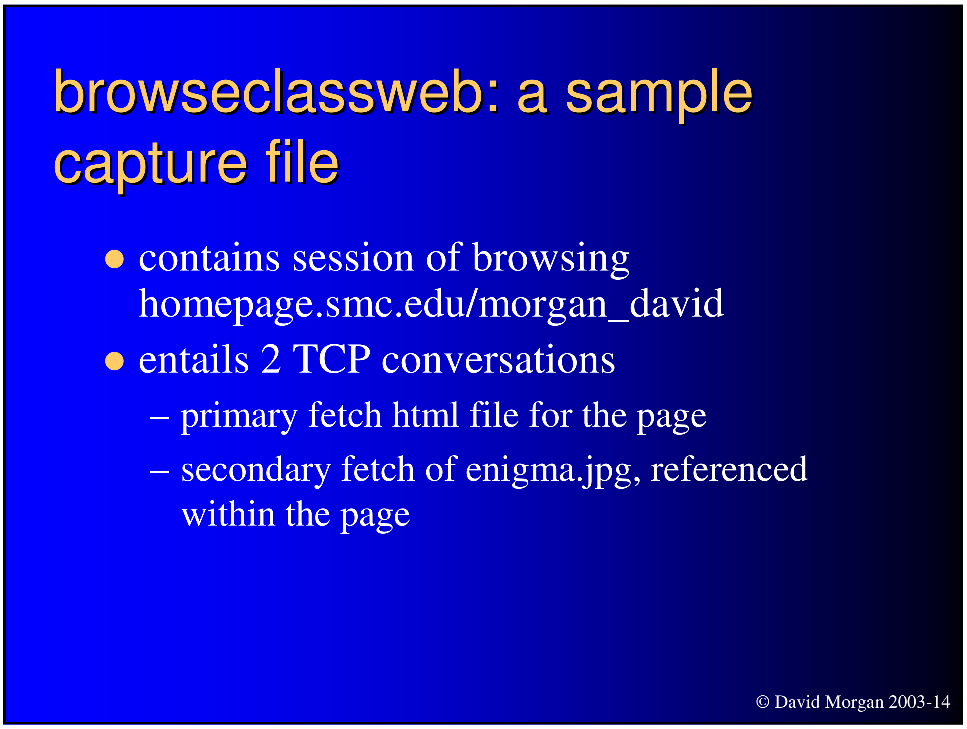
© David Morgan 2003-14



tcpslice

- extract dump file parcels by timestamp interval
- glue them together

© David Morgan 2003-14

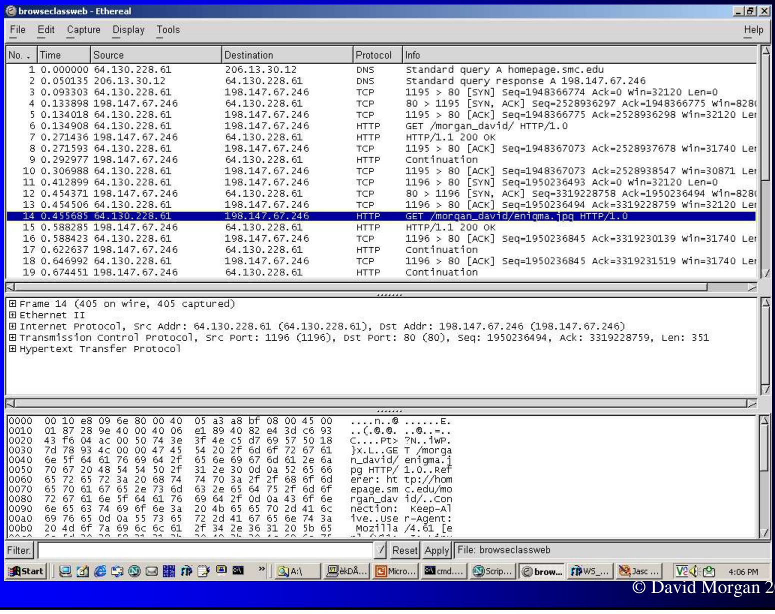


browseclassweb: a sample capture file

- contains session of browsing homepage.smc.edu/morgan_david
- entails 2 TCP conversations
  - primary fetch html file for the page
  - secondary fetch of enigma.jpg, referenced within the page

© David Morgan 2003-14

browseclassweb



browseclassweb - Ethereal
File Edit Capture Display Tools Help

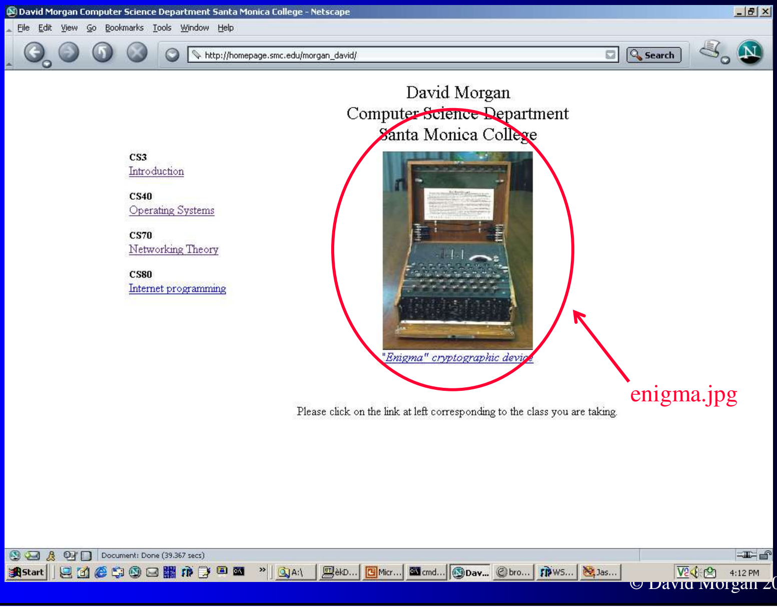
```
No. . Time     Source          Destination     Protocol Info
  1 0.000000 64.130.228.61   206.13.30.12    DNS      Standard query A homepage.smc.edu
  2 0.050135 206.13.30.12    64.130.228.61   DNS      Standard query response A 198.147.67.246
  3 0.093303 64.130.228.61   198.147.67.246  TCP      1195 > 80 [SYN] Seq=1948366774 Ack=0 Win=32120 Len=0
  4 0.133898 198.147.67.246  64.130.228.61   TCP      80 > 1195 [SYN, ACK] Seq=2528936297 Ack=1948366775 Win=828
  5 0.134018 64.130.228.61   198.147.67.246  TCP      1195 > 80 [ACK] Seq=1948366775 Ack=2528936298 Win=32120 Le
  6 0.134908 64.130.228.61   198.147.67.246  HTTP     GET /morgan_david/ HTTP/1.0
  7 0.271436 198.147.67.246  64.130.228.61   HTTP     HTTP/1.1 200 OK
  8 0.271593 64.130.228.61   198.147.67.246  TCP      1195 > 80 [ACK] Seq=1948367073 Ack=2528937678 Win=31740 Le
  9 0.292977 198.147.67.246  64.130.228.61   HTTP     Continuation
 10 0.306988 64.130.228.61   198.147.67.246  TCP      1195 > 80 [ACK] Seq=1948367073 Ack=2528938547 Win=30871 Le
 11 0.412899 64.130.228.61   198.147.67.246  TCP      1196 > 80 [SYN] Seq=1950236493 Ack=0 Win=32120 Len=0
 12 0.454371 198.147.67.246  64.130.228.61   TCP      80 > 1196 [SYN, ACK] Seq=3319228758 Ack=1950236494 Win=828
 13 0.454506 64.130.228.61   198.147.67.246  TCP      1196 > 80 [ACK] Seq=1950236494 Ack=3319228759 Win=32120 Le
 14 0.455685 64.130.228.61   198.147.67.246  HTTP     GET /morgan_david/enigma.jpg HTTP/1.0
 15 0.588285 198.147.67.246  64.130.228.61   HTTP     HTTP/1.1 200 OK
 16 0.588423 64.130.228.61   198.147.67.246  TCP      1196 > 80 [ACK] Seq=1950236845 Ack=3319230139 Win=31740 Le
 17 0.622637 198.147.67.246  64.130.228.61   HTTP     Continuation
 18 0.646992 64.130.228.61   198.147.67.246  TCP      1196 > 80 [ACK] Seq=1950236845 Ack=3319231519 Win=31740 Le
 19 0.674451 198.147.67.246  64.130.228.61   HTTP     Continuation
```

```
⊞ Frame 14 (405 on wire, 405 captured)
⊞ Ethernet II
⊞ Internet Protocol, Src Addr: 64.130.228.61 (64.130.228.61), Dst Addr: 198.147.67.246 (198.147.67.246)
⊞ Transmission Control Protocol, Src Port: 1196 (1196), Dst Port: 80 (80), Seq: 1950236494, Ack: 3319228759, Len: 351
⊞ Hypertext Transfer Protocol
```

```
0000  00 10 e8 09 6e 80 00 40  05 a3 a8 bf 08 00 45 00   ....n..@ ......E.
0010  01 87 28 9e 40 00 40 06  e1 89 40 82 e4 3d c6 93   ..(.@.@. ..@..=..
0020  43 f6 04 ac 00 50 74 3e  3f 4e c5 d7 69 57 50 18   C....Pt> ?N..iWP.
0030  7d 78 93 4c 00 00 47 45  54 20 2f 6d 6f 72 67 61   }x.L..GE T /morga
0040  6e 5f 64 61 76 69 64 2f  65 6e 69 67 6d 61 2e 6a   n_david/ enigma.j
0050  70 67 20 48 54 54 50 2f  31 2e 30 0d 0a 52 65 66   pg HTTP/ 1.0..Ref
0060  65 72 65 72 3a 20 68 74  74 70 3a 2f 2f 68 6f 6d   erer: ht tp://hom
0070  65 70 61 67 65 2e 73 6d  63 2e 65 64 75 2f 6d 6f   epage.sm c.edu/mo
0080  72 67 61 6e 5f 64 61 76  69 64 2f 0d 0a 43 6f 6e   rgan_dav id/..Con
0090  6e 65 63 74 69 6f 6e 3a  20 4b 65 65 70 2d 41 6c   nection:  Keep-Al
00a0  69 76 65 0d 0a 55 73 65  72 2d 41 67 65 6e 74 3a   ive..Use r-Agent:
00b0  20 4d 6f 7a 69 6c 6c 61  2f 34 2e 36 31 20 5b 65    Mozilla /4.61 [e
```

© David Morgan 2003-14

Target page



David Morgan Computer Science Department Santa Monica College - Netscape
File Edit View Go Bookmarks Tools Window Help
http://homepage.smc.edu/morgan_david/
Search
Document: Done (39.367 secs)
Start
A:\
èkD...
Micr...
cmd...
Dav...
bro...
WS...
Jas...
4:12 PM

David Morgan
Computer Science Department
Santa Monica College

CS3
Introduction

CS40
Operating Systems

CS70
Networking Theory

CS80
Internet programming

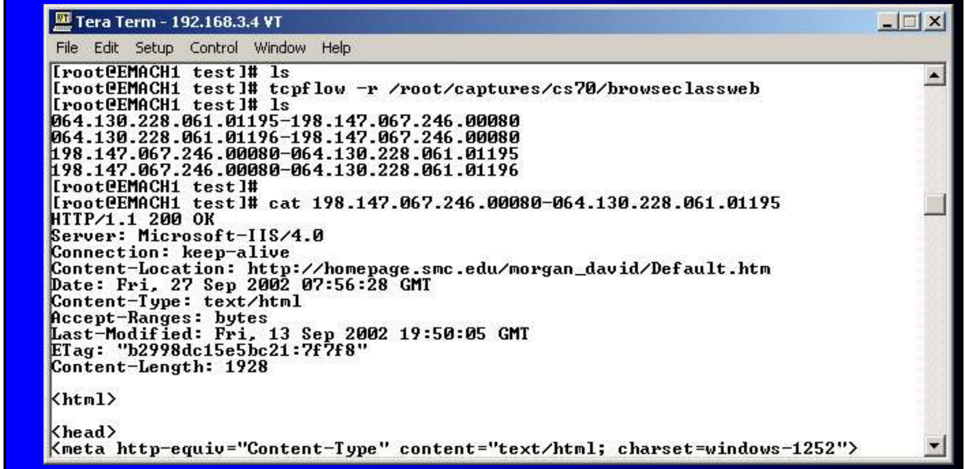
"Enigma" cryptographic device

Please click on the link at left corresponding to the class you are taking.

enigma.jpg

© David Morgan 2003-14

tcpflow


Tera Term - 192.168.3.4 VT
File Edit Setup Control Window Help

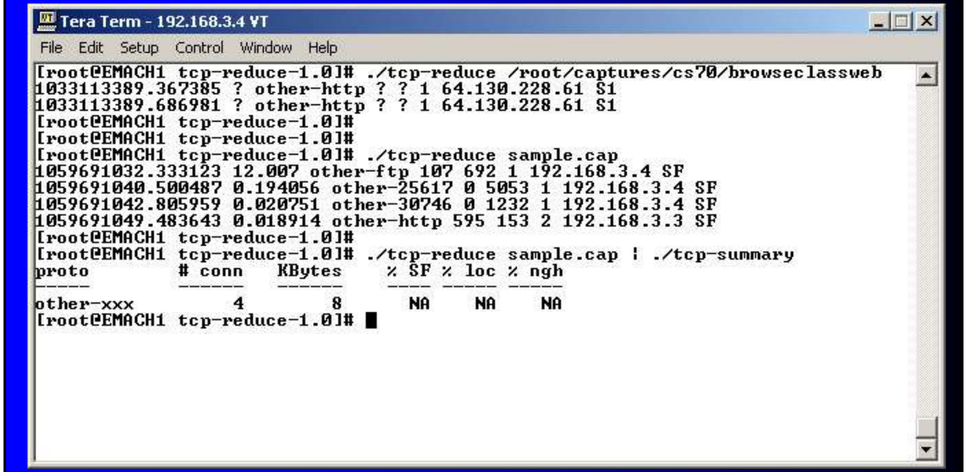
```
[root@EMACH1 test]# ls
[root@EMACH1 test]# tcpflow -r /root/captures/cs70/browseclassweb
[root@EMACH1 test]# ls
064.130.228.061.01195-198.147.067.246.00080
064.130.228.061.01196-198.147.067.246.00080
198.147.067.246.00080-064.130.228.061.01195
198.147.067.246.00080-064.130.228.061.01196
[root@EMACH1 test]#
[root@EMACH1 test]# cat 198.147.067.246.00080-064.130.228.061.01195
HTTP/1.1 200 OK
Server: Microsoft-IIS/4.0
Connection: keep-alive
Content-Location: http://homepage.smc.edu/morgan_david/Default.htm
Date: Fri, 27 Sep 2002 07:56:28 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Fri, 13 Sep 2002 19:50:05 GMT
ETag: "b2998dc15e5bc21:7f7f8"
Content-Length: 1928

<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
```

© David Morgan 2003-14

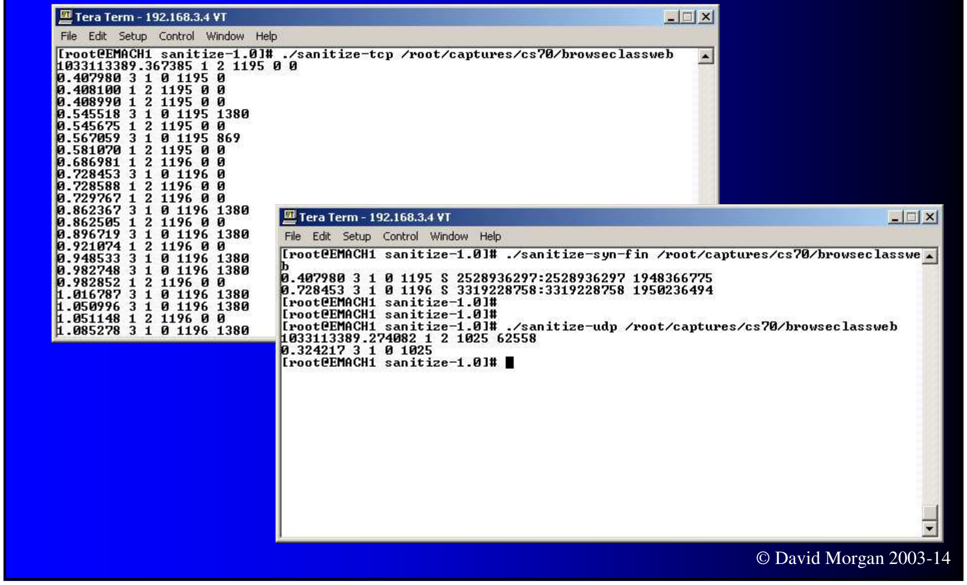
tcp-reduce


Tera Term - 192.168.3.4 VT
File Edit Setup Control Window Help

```
[root@EMACH1 tcp-reduce-1.0]# ./tcp-reduce /root/captures/cs70/browseclassweb
1033113389.367385 ? other-http ? ? 1 64.130.228.61 S1
1033113389.686981 ? other-http ? ? 1 64.130.228.61 S1
[root@EMACH1 tcp-reduce-1.0]#
[root@EMACH1 tcp-reduce-1.0]#
[root@EMACH1 tcp-reduce-1.0]# ./tcp-reduce sample.cap
1059691032.333123 12.007 other-ftp 107 692 1 192.168.3.4 SF
1059691040.500487 0.194056 other-25617 0 5053 1 192.168.3.4 SF
1059691042.805959 0.020751 other-30746 0 1232 1 192.168.3.4 SF
1059691049.483643 0.018914 other-http 595 153 2 192.168.3.3 SF
[root@EMACH1 tcp-reduce-1.0]#
[root@EMACH1 tcp-reduce-1.0]# ./tcp-reduce sample.cap ¦ ./tcp-summary
proto         # conn    KBytes     % SF % loc % ngh
-----         ------    ------     ---- ----- -----
other-xxx          4         8       NA    NA    NA
[root@EMACH1 tcp-reduce-1.0]# █
```

© David Morgan 2003-14

sanitize



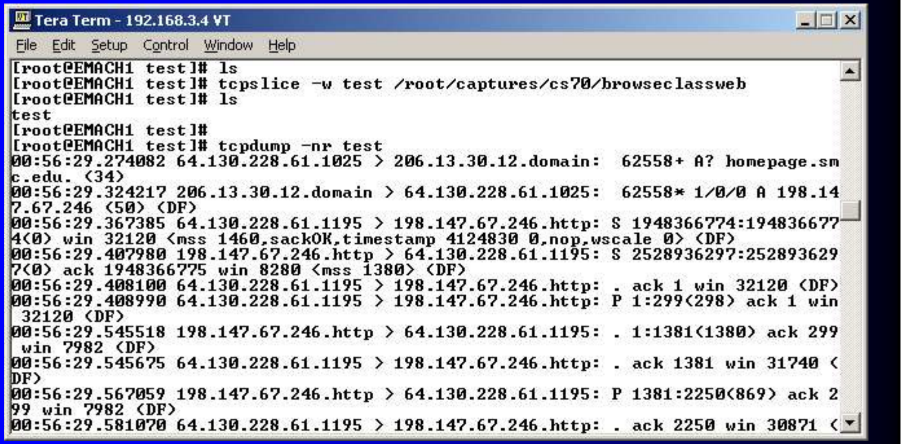
Tera Term - 192.168.3.4 VT
File Edit Setup Control Window Help
Tera Term - 192.168.3.4 VT
File Edit Setup Control Window Help

```
[root@EMACH1 sanitize-1.0]# ./sanitize-tcp /root/captures/cs70/browseclassweb
1033113389.367385 1 2 1195 0 0
0.407980 3 1 0 1195 0
0.408100 1 2 1195 0 0
0.408990 1 2 1195 0 0
0.545518 3 1 0 1195 1380
0.545675 1 2 1195 0 0
0.567059 3 1 0 1195 869
0.581070 1 2 1195 0 0
0.686981 1 2 1196 0 0
0.728453 3 1 0 1196 0
0.728588 1 2 1196 0 0
0.729767 1 2 1196 0 0
0.862367 3 1 0 1196 1380
0.862505 1 2 1196 0 0
0.896719 3 1 0 1196 1380
0.921074 1 2 1196 0 0
0.948533 3 1 0 1196 1380
0.982748 3 1 0 1196 1380
0.982852 1 2 1196 0 0
1.016787 3 1 0 1196 1380
1.050996 3 1 0 1196 1380
1.051148 1 2 1196 0 0
1.085278 3 1 0 1196 1380
```

```
[root@EMACH1 sanitize-1.0]# ./sanitize-syn-fin /root/captures/cs70/browseclasswe
b
0.407980 3 1 0 1195 S 2528936297:2528936297 1948366775
0.728453 3 1 0 1196 S 3319228758:3319228758 1950236494
[root@EMACH1 sanitize-1.0]#
[root@EMACH1 sanitize-1.0]#
[root@EMACH1 sanitize-1.0]# ./sanitize-udp /root/captures/cs70/browseclassweb
1033113389.274082 1 2 1025 62558
0.324217 3 1 0 1025
[root@EMACH1 sanitize-1.0]# █
```

© David Morgan 2003-14

tcpslice



Tera Term - 192.168.3.4 VT
File Edit Setup Control Window Help

```
[root@EMACH1 test]# ls
[root@EMACH1 test]# tcpslice -w test /root/captures/cs70/browseclassweb
[root@EMACH1 test]# ls
test
[root@EMACH1 test]#
[root@EMACH1 test]# tcpdump -nr test
00:56:29.274082 64.130.228.61.1025 > 206.13.30.12.domain:  62558+ A? homepage.sm
c.edu. (34)
00:56:29.324217 206.13.30.12.domain > 64.130.228.61.1025:  62558* 1/0/0 A 198.14
7.67.246 (50) (DF)
00:56:29.367385 64.130.228.61.1195 > 198.147.67.246.http: S 1948366774:194836677
4(0) win 32120 <mss 1460,sackOK,timestamp 4124830 0,nop,wscale 0> (DF)
00:56:29.407980 198.147.67.246.http > 64.130.228.61.1195: S 2528936297:252893629
7(0) ack 1948366775 win 8280 <mss 1380> (DF)
00:56:29.408100 64.130.228.61.1195 > 198.147.67.246.http: . ack 1 win 32120 (DF)
00:56:29.408990 64.130.228.61.1195 > 198.147.67.246.http: P 1:299(298) ack 1 win
 32120 (DF)
00:56:29.545518 198.147.67.246.http > 64.130.228.61.1195: . 1:1381(1380) ack 299
 win 7982 (DF)
00:56:29.545675 64.130.228.61.1195 > 198.147.67.246.http: . ack 1381 win 31740 (
DF)
00:56:29.567059 198.147.67.246.http > 64.130.228.61.1195: P 1381:2250(869) ack 2
99 win 7982 (DF)
00:56:29.581070 64.130.228.61.1195 > 198.147.67.246.http: . ack 2250 win 30871 (
```

© David Morgan 2003-14