Open source tools

Dr.G.Usha Devi
Associate Professor
School of Information Technology and Engineering
VIT University
Vellore

MRTG

MRTG

- Multi Router Traffic Grapher
- produces Web pages that display graphs of bandwidth use on network links on daily, weekly, monthly, and yearly scales.
- invaluable tool for diagnosing network problems because it not only indicates the current status of the network but also visually compare this with the history of network utilization.

- MRTG relies on SNMP version one, and optionally SNMP version two,
- to obtain data from routers or other network hardware.
- MRTG sends SNMP requests every five minutes and stores the responses in a specialized data format.
- The graphs are created in Portable Network Graphics (PNG) format and can be included in Web pages or used in other
- MRTG is useful for studying trends in traffic on your network

Installing MRTG

- http://www.mrtg.org/
- it requires:
  - Perl 5.005 or greater
  - The GD library
  - The PNG library
  - The zlib library

1. gunzip -c mrtg-2.9.25.tar.gz | tar xvf –
2. cd mrtg-2.9.25
3. ./configure
4. make
5. make install

the default location, /usr/local/mrtg-2/

Building the PNG Library

- http://www.libpng.org/pub/png/libpng.html

1. gunzip -c libpng-1.2.5.tar.gz | tar xvf –
2. cd libpng-1.2.5
3. cp scripts/makefile.linux makefile

   examine the INSTALL file. It contains a list of makefiles designed for use with different systems. For example

   makefile.solaris

   makefile.hpux

   makefile.macosx

4. make
5. make install

Building the GD Library

- http://www.boutell.com/gd/

1. gunzip -c gd-2.0.9.tar.gz | tar xvf –
2. cd gd-2.0.9
3. Before running the configure script, indicate where GD should find the PNG library by setting the CFLAGS and LDFLAGS environment variables.
    1. CFLAGS=-I/usr/local/include export CFLAGS
    2. LDFLAGS="-L/usr/local/lib -R/usr/local/lib" export LDFLAGS
4. make install

Building MRTG

1. Change back to the MRTG source directory and now run the configure script with CFLAGS and LDFLAGS still pointed at /usr/local:
    1. CFLAGS=-I/usr/local/include export CFLAGS
    2. LDFLAGS="-L/usr/local/lib -R/usr/local/lib" export \ LDFLAGS
2. ./configure
3. make
4. make install

This will install all of the MRTG software in /usr/local/mrtg-2.

Configuring MRTG

- Once installed, you need to configure it to monitor your devices.
  - includes generating the config file
  - generating HTML index pages for the graphs and
  - setting up MRTG to gather data at regular intervals

Generating the Configuration File

- the configuration file is in /usr/local/mrtg-2/cfg/mrtg.cfg
- If the directory does not exist yet, so we create it,
  - mkdir /usr/local/mrtg-2/cfg
  - chmod 700 /usr/local/mrtg-2/cfg
-

- Now run the cfgmaker program to create the configuration file:

```
/usr/local/mrtg-2/bin/cfgmaker \
--global 'WorkDir: /usr/local/apache/htdocs/mrtg' \
--global 'Options[_]: bits' \
--global 'IconDir: icons' \
--snmp-options=:::::2 \
--subdirs=HOSTNAME \
--ifref=ip \
--ifdesc=alias \
--output /usr/local/mrtg-2/cfg/mrtg.cfg \
community@router1.example.com \
community@router2.example.com \
community@router3.example.com
```

It will spend a short while probing each device in order to build the configuration

- Copy the icons from the MRTG distribution to this directory now:

mkdir /usr/local/apache/htdocs/mrtg/icons

cp /usr/local/mrtg-2/share/mrtg2/icons/* \ /usr/local/apache/htdocs/mrtg/icons/

and make sure the directory and files are readable to your Web server.

Generating Initial Data

- Once the configuration file has been created, you can run the mrtg program.

  /usr/local/mrtg-2/bin/mrtg /usr/local/mrtg-2/cfg/mrtg.cfg
- After running the mrtg command, you will be able to find PNG files and HTML files in /usr/local/apache/htdocs/mrtg.
- A typical Apache Web server installation will allow you to view the pages at http://server.example.com/mrtg, where server.example.com is replaced with the name of the machine you are installing MRTG on.

Using MRTG

- **Faulty Data**
- **Missing Data**

Maintaining MRTG

- MRTG requires more maintenance
- Each time you move a network or router interface, you will have to make sure the change is reflected in the MRTG configuration.
- In the event that you do make a change that causes MRTG to lose its sense of which data belongs to which network, you can attempt to remedy the situation by finding the appropriate .log file under /usr/local/apache/htdocs/mrtg/router* and renaming it to be the data file that MRTG expects for the new network.

Neo

Overview

- A tool is needed that handles all of the SNMP requests
- check bandwidth usage or
- determine on which switch port a particular host resides.
- Neo is a text-based client
- Tool to use a remote login session to manage the network.
- Neo is helpful only if your devices can be managed with SNMP.

What Neo Can Help You Do

- Determining on which switch port a host resides
- Translating an IP address to a hardware address
- Obtaining traffic statistics
- Obtaining board or port information
- Disabling or enabling a port
- Obtaining general device information
- Obtaining information about device power and environmental conditions

Locating Hosts and the Forwarding Table

- You can find the port on which a particular hardware address is located:

neo: locate 00:03:BA:09:1F:36 @switch13.example.com

Found on 6@switch13.example.com

- Or ask the switch for all hardware addresses present on a particular port:

neo: port search 2/5@switch2.example.com

00:04:76:CB:B1:CD

00:05:02:4B:C6:50

00:06:5B:1D:68:43

00:06:5B:1D:68:46

00:50:8B:AD:FE:8E

00:E0:63:2B:D7:C4

00:E0:63:2B:D7:DC

Translating an IP Address to a Hardware Address

- Neo can use the SNMP ipNetToMediaTable to ask a device for the hardware address associated with a particular IP address:

neo: arpfind host.example.com

router.example.com 10.5.0.1 says 10.5.1.2 is 00:03:BA:09:1F:36

Obtaining Traffic Statistics

- **neo: stats switch.example.com**

  Probing devices ...

  Getting first set of stats...

  Getting second set of stats...

  Port statistics:

p type u lnk adm ap kbs ikbs okbs pps ipps opps ierps oerps

---

1 100TX 100 On 20 0 20 26 0 26 0 0

2 100T 100 On 19 0 19 26 0 26 0 0

Disabling or Enabling a Port

- **neo: port disable 22@switch.example.com**
  22@switch.example.com disabled
- **neo: port enable 22@switch.example.com**
  22@switch.example.com enabled

Installing Neo

- http://web.mit.edu/ktools/

1. gunzip -c neo-1.2.12.tar.gz | tar xvf –
2. cd neo-1.2.12
3. ./configure
4. make
5. make install

the Neo binary in /usr/local/bin/.

- Neo builds its own small SNMP implementation and therefore does not require any external SNMP packages.
- Once the binary is built, try running it to make sure it works.

  ./neo

  Welcome to neo version 1.2.12 (Atropine).
  Consult 'help version' for information on features in this version.

  neo:

If so, you have successfully built Neo and are ready to begin using it.

Using Neo

- **The Command Prompt**
  - **Basic Readline Editing Keys.**

Key	Function
C-f / Right	Move cursor right
C-b / Left	Move cursor left
C-a	Move cursor to start of line
C-e	Move cursor to end of line
M-f	Move cursor to next word
M-b	Move cursor to previous word
C-p / Up	Retrieve the previous command
C-n / Down	Retrieve the next command
C-k	Kill text to end of line
C-y	Yank killed text into buffer

The Location Syntax

Location	Meaning
@switch1	The device switch1
Switch1	The device switch1
@switch1,switch5	The devices switch1 and switch5
@f:/var/tmp/devicelist	The devices listed in /var/tmp/devicelist
@k:east	Devices associated with "east" in the keyfile
3/2@switch5	Board 3, port 2 on switch5
10@switch1	Port 10 on switch1
3/@switch5	Board 3 on switch5
*@switch1	All ports on switch1
.@switch1	All ports on switch1
3/*@switch5	All ports on board 3 on switch5
/@switch5	All ports on switch5
*/@switch5	All boards on switch5

Variables

- Neo has a number of built-in variables it uses for controlling user configuration options.
- **neo: print**

writecom = 'public'

readcom = 'public'

keyfile = '/usr/local/etc/neo.keyfile'

statsdelay = 5

timeout = 8

macmode = 'standard'

version = '1.2.12 (Atropine)'

burst = 1

The Arpfind Command

neo: arpfind myhost.example.com router.example.com

router.example.com says 10.5.1.2 is 00:03:BA:09:1F:36

The Locate Command

neo: locate 00:03:BA:09:1F:36 @k:northannex

Found on 6@switch13.example.com

The Port Command

- port enable: Turn the port on
- port disable: Turn the port off
- port status: Report the administrative status of the port
- port search: Print the address of all devices on the port

The Device Summary Command

- device summary
  - neo: dev sum switch1.example.com
- device info
  - neo: dev info switch.example.com
    switch.example.com
    Device type: C2200
    Contact : admin@example.com
    Name : switch.example.com
    Location : 5-142T
    Uptime : 92 days 22:07:20
    ObjectID : .1.3.6.1.4.1.52.3.9.3.4.84
    Descr : Cabletron Systems, Inc. 2H253-25R Rev 04.00....
- device type
  - neo: stats *@switch1.example.com
    Probing devices ... Getting first set of stats... Getting second set of stats...
    Port statistics:

Maintaining Neo

- Neo requires two maintenance tasks.
  - regularly updating the installed version of Neo simply because, as with any new piece of software, features and bug fixes are constantly being added. The latest version is always available on the Web site.
  - updating the keyfile, if you use one. Neo is an effective tool only if it knows which devices to contact. Make sure your keyfile reflects the current state of your network either by updating it manually or by using an automated process.
- Neo does not require you to maintain SNMP MIBs; any necessary variable data is coded into the program itself.

NetFlow

Overview of NetFlow and Flow-Tools

- a more qualitative view of traffic.
  - If flooding of packets, to know something about the data in the packets.
- Allow to view the information such as the source and destination IP addresses, source and destination protocol port numbers, number of packets transmitted, number of bytes transmitted, and much more.
- An excellent set of open source tools for collecting and processing NetFlow data is the Flow-Tools package.
  - It includes utilities for collecting flows on a server, storing the results, and printing and manipulating flows as well as tools for producing reports based on the data.

How NetFlow Works

- **Flows**
- Flow is what you would naturally think of as one full network conversation.
- Examples :
  - The Transmission Control Protocol (TCP) connection a workstation opens to retrieve a Web page is a flow.
    - The start of the TCP connection is the beginning of the flow, and when the connection is closed, the flow is complete.
  - the series of ICMP packets that make up a ping test. The flow begins with the first ICMP packet and ends with the last.
- Every flow must have a unique set of the following:
  - Source IP address
  - Destination IP address
  - Source port
  - Destination port
  - IP protocol number
  - Type of service field
  - Input interface
- a flow is unidirectional

NetFlow Versions

- In Version 1, each flow contained the following information:
  - Source IP address
  - Destination IP address
  - Source port
  - Destination port
  - Next hop router address
  - Input interface number
  - Output interface number
  - Number of packets sent
  - Number of bytes sent
  - sysUpTime when flow began
  - sysUpTime when flow ended
  - IP protocol number
  - IP type of service
  - Logical OR of all TCP flags seen

- Version 5.
  - Includes Autonomous System (AS) numbers from Border Gateway Protocol (BGP) if available and
  - also includes sequence numbers, which allows checking for lost packets.
- Version 7.
  - to perform NetFlow accounting.
- Version 8.
  - Introduces Router-Based NetFlow Aggregation, a feature that can greatly reduce the amount of data sent when flows are exported from the router.
- Version 9.
  - Introduces a template-based scheme for reporting flows, which allows a client receiving exported flow data to process the data without necessarily having prior knowledge of what kinds of data will be present.

Installing Flow-Tools

- http://www.net.ohiostate.edu/software/
- gunzip -c flow-tools-0.59.tar.gz | tar xvf –
- cd flow-tools-0.59
- ./configure
- make
- make install

the programs will be in /usr/local/netflow/bin/

Configuring NetFlow on the Router

- configure the router to export flows to a host that will be running the Flow-Tools software.
- On a Cisco router, you can enable NetFlow on an interface with the interface configuration command ip route-cache flow. For example:
- router#config term
- Enter configuration commands, one per line. End with CNTL/Z.
  - router(config)#int Ethernet1/2
  - router(config-if)#ip route-cache flow
  - router(config-if)#end
- To instruct the router to export flows, use the ip flow-export destination configure command:
  - router(config)#ip flow-export destination 192.0.2.5 9995
  - router(config)#ip flow-export source Loopback0
  - router(config)#ip flow-export version 5

Using Flow-Tools

- A number of programs make up the Flow-Tools software package.

Programs Included with Flow-Tools

Program	Function
flow-receive	Receive flows, send to stdout
flow-capture	Receive flows, store to disk
flow-print	Print flow data to the screen
flow-report	Produce flow reports for other programs
flow-stat	Print flow statistics to the screen
flow-dscan	Detect suspicious network traffic
flow-cat	Concatenate flow data files
flow-merge	Sort and concatenate flow data files
flow-expire	Remove old flow data files
flow-split	Split data files into smaller files
flow-header	Print meta data on a capture session
flow-fanout	Redistribute UDP exports to other addresses
flow-send	Send flow data in NetFlow format
flow-tag	Add user-defined tags to flows
flow-filter	Filter flows
flow-gen	Generate test flow data
flow-import	Import data from other NetFlow tools
flow-export	Export data to other NetFlow tools
flow-xlate	Translate flow data

Capturing Flows

- flow-capture
  - which not only receives the flows, but also stores them to files on disk, rotates the files, and ensures that they do not grow larger than a limit you specify on the command line.
- flow-receive
  - which sends the flow data to the standard output instead of storing flows on disk

Example

- Linux# flow-receive 0/0/9995 | flow-print

```
flow-receive: setsockopt(size=262144)
flow-receive: New exporter: time=1048976578 src_ip=10.255.255...
flow-receive: New exporter: time=1048976578 src_ip=10.255.255...
flow-receive: New exporter: time=1048976578 src_ip=10.255.255...
srcIP            dstIP         prot  srcPort   dstPort   octets   packets
10.26.196.41   10.221.0.157   17     2706       1497       35        1
```

Options to the Flow-Capture Program

Flag	Function
-A	Replace 0 AS values
-b	Specify output byte order
-c	Enable TCP client connections
-C	Add a comment
-d	Turn on debugging
-e	Maximum number of files
-E	Maximum data size to store
-f	File name of filter list
-F	Specify active filter
-h	Display help
-m	Use a privacy mask
-n	Specify number of new files per day
-N	Set nesting level
-p	Specify PID file
-R	Execute a program after rotation
-S	Syslog timestamps at specified interval
-t	File to load tags from
-T	Tags to use
-V	Specify output format
-w	Specify directory to store data files in
-z	Set compression level

- **Bounding the Data Size**
  - # flow-capture -w /var/tmp/flows -E 1G 0/0/9995
- **Changing Directory Nesting**
  - # find /var/tmp/flows
  - /var/tmp/flows
  - /var/tmp/flows/2003
  - /var/tmp/flows/2003/2003-03
- **Changing the File Rotation Rate**
  - #flow-capture -w /var/tmp/flows -n 23 0/0/9995
- **Changing the Process ID File**
  - # flow-capture -w /var/tmp/flows -p /var/tmp/pid 0/0/9995
- **Using Compression**
  - flow-capture -w /var/tmp/flows -z0
- **Killing Flow-Capture**
  - kill -QUIT 'cat /var/run/flow-capture.pid.9995‘
- **Allowing Remote Clients**
  - flow-capture -w /var/tmp/flows -c 10 0/0/9995
    Client% nc server.example.com 9995 | flow-print

srcIP dstIP prot srcPort dstPort octets packets

Viewing Flow Data

- Flow-print.
  - Prints flow data in ASCII.
- Flow-report.
- Flow-stat.
  - generates reports based on statistics such as high bandwidth use by port.
- Flow-dscan.
  - Analyzes flow data to find suspicious network traffic such as port scans.

Manipulating Flow Data

- **Flow-Cat and Flow-Merge**

```
# flow-merge ft-v01.2003-03-30.140900-0500 \
                ft-v01.2003-03-30.140937-0500 | flow-print
```

- **Flow-Split**
- **Flow-Expire**
- **Flow-Header**

OAK

Overview of Oak

- many network devices such as switches and routers, keep a log of messages about operational conditions.
- This message log is a useful tool for analyzing a problem after it has occurred.
- When an interface on a router stops functioning, you can login to the router and examine the error log for information about the problem.
- Drawbacks
  - a system that requires to login to a device in order to access its system logs
  - It is also difficult to monitor the log files via a login session if you have a large number of devices in your environment.

- For this reason, most devices that keep a local message log are also capable of sending notifications to a remote host using the **syslog protocol**.
- Using the syslog protocol, you can configure all of your servers and network devices to send error messages to a single machine, and from that machine, you can monitor the resulting error log.
- If a critical problem is detected, an operator should be notified immediately.
- Less serious problems can be reported in an hourly or daily message.
- Oak is a program developed at MIT that will **process messages and allow you to respond appropriately**

What Oak Can Help You Do

- Oak examines a message log in syslog format and allows you to:
  - Ignore unimportant messages
  - Condense redundant information
  - Produce reports of important messages
  - Notify operators immediately of critical messages

- The Oak configuration file will specify which messages are important to you and how you wish to be notified in the event they should be received.
- Example
  - Oak configured to send a daily report in email, an hourly report in an instant message to the operational group, and an immediate instant message to the operational group if a critical problem is detected.

Hourly message log
SERVER1.EXAMPLE.COM:
2: login: ROOT LOGIN console
1: syslogd: going down on signal 15
1: Use is subject to license terms. ** Too many messages found for host, truncating **

ROUTER.EXAMPLE.COM:
6: ___:___ %LINEPROTO-5-UPDOWN: \ Line protocol on Interface Ethernet9/4, change$
5: ___:___ %LINEPROTO-5-UPDOWN: \ Line protocol on Interface Ethernet9/4, change$
2: ___:___ %LINEPROTO-5-UPDOWN: \ Line protocol on Interface Ethernet9/2, change$
1: ___:___ %LINEPROTO-5-UPDOWN: \ Line protocol on Interface Ethernet9/2, change$

- A **time-critical message** might look like this:

**** CRITICAL MESSAGE LOG ****
SERVER4.EXAMPLE.COM:
ufs: NOTICE: alloc: /var: file system full

server with a full filesystem, which should be reported to an administrator

Installing Oak

- http://web.mit.edu/ktools/

1. gunzip -c oak-1.3.5.tar.gz | tar xvf –
2. cd oak-1.3.5
3. ./configure
4. make
5. make install

the oak binary is in /usr/local/bin/

Using Oak

- Before configuring Oak to notify you of system events, you must first configure your servers and network devices to forward their syslogs to a central server
- **Configuring Syslog on Unix Workstations**
  - Every syslog message has four basic parts:
    - The system facility
    - A severity level
    - A time stamp
    - The message content
  - **Facility Types**
    - user
    - Kern - messages from the kernel
    - Mail - messages about the mail system
    - daemon
    - auth
    - lpr
    - News
    - local0-7 - eight facilities reserved for local admins

Severity Level Values

- each syslog message has a level of severity
  - emerg -most severe -immediate attention is required
  - alert
  - crit
  - err
  - warning
  - notice
  - Info
  - debug - least severe.
- Purpose:
  - is to allow operators to sort syslog messages by priority

SYSLOG CONFIGURATION FILE

- is at /etc/syslog.conf
- Format:

Facility.severity level	**filename** (where to send the matching messages)
*.err;kern.notice;auth.notice	/dev/console
*.err;kern.debug;daemon.notice	/var/adm/messages
*.emerg	*
auth.info	/var/log/auth
auth.notice	/dev/console
mail.info	/var/log/mailer
daemon.info	/var/log/daemon
local2.notice	/var/log/inetd

- **Syslog Actions**

Action	**Syntax**
Append to file	filename beginning with slash
Send to logged-in user	username
Send to logged-in users	user1, user2, ...
Send to all logged-in users	*
Send to a remote host	@hostname

- ps -ef | grep syslog

root 211 1 0 Sep 19 ? 0:17 /usr/sbin/syslogd

- kill -HUP 211

send a test syslog message using the logger program

- logger -pwarn "This is a test“

send a message to syslog at the user.warn level

Configuring Syslog on Network Devices

- Every device uses a different syntax for configuring remote logging.
- Cisco IOS uses the logging command from configure mode:
  - Router(config)#logging 10.7.21.88
  - send log messages to the logging host 10.7.21.88 and default facility will be local0
  - Router(config)#logging facility local3
- On Cisco devices running CatOS
  - switch18> (enable) set logging server 10.7.21.88
  - switch18> (enable) set logging server enable

An Introduction to Regular Expressions

- an integral part of the Oak configuration language
- syntax used to represent a pattern that a text string can either match or not match
- For example
  - grep **domain** /etc/resolv.conf
    domain EXAMPLE.COM
  - grep do..in /etc/resolv.conf
  - grep do.in /etc/resolv.conf

(.+) to replace the characters between the brackets because it is redundant information

Characters	Description	Example
.	any character except a newline character	foo.*bar
*	Zero or more times	fo*bar
+	One or more times	foo[123]+bar
^	Beginning of line	^domain
$	End of line	foobar$
[ ]	Any one of the characters in the bracket	foo[123]bar [0-9]
\	actual character should be matched	\.
( )	to grab a section of text for later use	foo(.+)bar Matches foo52bar string "52" will be stored in a variable that can be used later

Configuring Oak

- based on the idea of a message queue
- Each queue is defined to take a certain action at a specified time interval
- One queue may send an email every morning
- Another may send an instant message each hour
- After the queues are defined, you will define regular expressions that control which messages are sent to which queues
- The typical Oak configuration follows this order:
  - Set global options
  - Define queues
  - Define regular expressions for critical messages
  - Define regular expressions for trash messages
  - Define regular expressions for summarizing other messages
  - Define a catch-all regular expression for everything else

1. Global Options

- **Facility Types**

Option	Function
set infile file	Define the file to be monitored
set nukepid	Automatically remove process IDs
set no nuke pid	Do not automatically remove PIDs
set nukeciscoid	Remove log IDs from cisco syslogs
set no nukeciscoid	Do not remove Cisco log IDs
set nukesmqid	Remove Sendmail queue IDs
set no nukesmqid	Do not remove Sendmail queue IDs
set ignorehosts host [host ...]	Ignore logs from the listed hosts
set onlyhosts host [host ...]	Process logs only from the listed hosts
set replacestr string	Replace text with string instead of underscores

- Examples
  - set infile /var/adm/messages
  - set nukepid
  - set nukeciscoid
  - set nukesmqid

2. Defining Queues

- Format : define queue queuename
- Example : sends an email message every day at 9:00 a.m

```
define queue network-gazette
prescan
action mail admin@example.com devnull@example.com "Daily Report"
action-limits 1000 100 100 100
fire 09:00
header  Daily message log
```

- prescan option will pick up messages already in the system log
- Action command
  - Mail - for email
  - zephyr - instant messaging system
  - exec - to run any external program
- action-limits
  - Maximum number of lines
  - Maximum number of characters on a line
  - Maximum number of hosts to report on
  - Maximum number of logs per host
- locking command
  - to suppress repeated notifications from this queue for a certain period of time
  - Example : locking 30m

3. Defining Regular Expressions

- It matches messages from sendmail when it complains about being unable to fork

```
# critical messages
on ^sendmail\[(.+)\]: (.+): SYSERR.*: (.+): cannot fork:
    queues testqueue
```

- Trash queue

```
# trash
on ^(.+):(.*)%SYS-5-CONFIG_I: Configured from console
    queues trash
```

- Finally, at the very end, you may include a statement catching anything that has not already been matched

```
on .*
    queues network-zephyr network-gazette
```

Running Oak

- Oak runs as a daemon and is invoked simply as
  - oak -c configfile

- Oak does not necessarily need to be run with root privileges. It does need to have access to read the syslog file it monitors, however.

- Check that the account you will run Oak from has access to read the syslog file.

A Small Sample Configuration

```
# global options
set infile /var/adm/messages
# define queues
define queue testqueue
     action mail admin@example.com devnull@example.com Report
               action-limits 1000 100 100 100
     fire *5m
     header 5 Minute Test Report
# critical messages
on ^sendmail\[(.+)\]: (.+): SYSERR.*: (.+): cannot fork:
     queues testqueue
# trash
on ^(.+):(.*)%SYS-5-CONFIG_I: Configured from console
     queues trash
# other
on ^(.+): %SYS-3-CPUHOG: Task ran for (.+) msec \((.+)\),
     queues network-zephyr network-gazette
on .*
     queues network-zephyr network-gazette
```

Maintaining Oak

- creating and updating the configuration file
- The Oak configuration must be updated to reflect the changes, if you change other components of your system, such as upgrading router software or deploying new servers
- It is good practice to update the configuration every few weeks

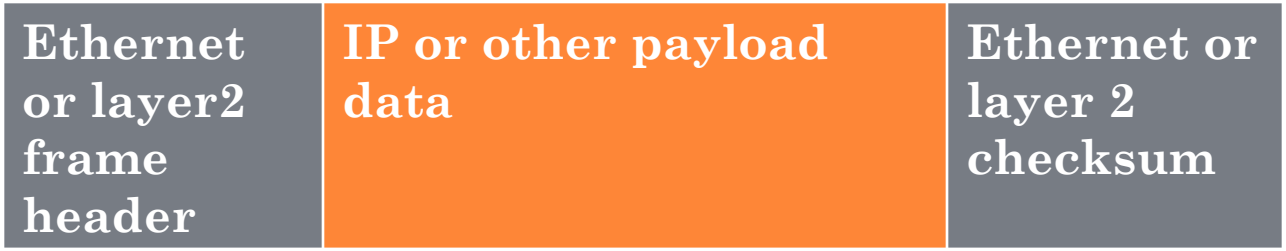
TCPDUMP

OVERVIEW

- open source tool for directly analyzing packets is a program called tcpdump
- http://www.tcpdump.org/
- there are still many protocols that transport data unencrypted
- When using a packet analyzer to monitor network traffic, you will be able to view private data sent by users on the network
- There are serious legal implications to monitoring such encrypted data because it can be considered a form of wire tapping

What Tcpdump Can Help You Do

- Tcpdump will allow you to view the entire data portion of an Ethernet frame or other link layer protocol and can optionally print the frame header as well



An Ethernet or Layer 2 Frame

- In common use this means tcpdump will allow you to view the entirety of an IP packet, an ARP packet, or any protocol at a higher layer than Ethernet. By default, tcpdump prints packets at the IP layer.

- **typical tcpdump output**
- 11:51:46.637811 10.25.71.241.80 > 10.18.0.100.61965: . ack 415 ...
  11:51:46.643077 10.25.71.241.80 > 10.18.0.100.61966: . ack 415 ...
  11:51:46.644830 10.209.29.151.80 > 10.18.0.100.61961: . ack 458...
  11:51:46.653025 10.18.0.100 > 10.7.14.114: icmp: echo request (DF)
  11:51:46.653226 10.7.14.114 > 10.18.0.100: icmp: echo reply (DF)
  11:51:46.658675 10.209.29.137.53 > 10.18.0.100.53454: 46268*- 2...
  11:51:46.659970 10.18.0.100.53454 > 10.70.10.79.53: 23134 A? sn...
  11:52:24.306670 arp who-has 10.18.1.80 tell 10.18.0.1
- **tcpdump to print all the data within each packet**

  16:05:52.209620 10.7.21.77.80 > 10.18.0.100.62532: P 1:236(235)...

  4500 0113 27a4 4000 3f06 d977 0a07 154d

  0a12 0064 0050 f444 dec4 4cd8 5894 b1d4

  .....

  0a0d 0a

- Tcpdump can also help debug denial of service attacks
- If a network is flooded and all other attempts to determine the source or destination of the traffic fail, tcpdump will show you the source address, destination address, and type of traffic involved
- useful for examining the contents of the traffic and the nature of the attack.
- **Limitations of Tcpdump**
  - a typical Ethernet card will **discard packets with an invalid checksum**. Therefore, tcpdump will not be a helpful tool for detecting this kind of broken packet on your network. For that, you will need specialized hardware.
  - Tcpdump is also able to report on only what it finds in the packet. If an **IP address is forged** in the packet, tcpdump has no ability to report anything else. Be aware that tcpdump is showing you only what the data is, not what it ought to be.

Installing Tcpdump

- Installed already. /usr/sbin/tcpdump
  - To check that,
    - #type tcpdump
- http://www.tcpdump.org/
- **The Pcap Library**
  - uncompress libpcap.tar.Z
  - tar xvf libpcap.tar
  - cd libpcap-0.4
  - ./configure
  - make
  - make install
- **Tcpdump**
  - uncompress tcpdump.tar.Z
  - tar xvf tcpdump.tar
  - cd tcpdump-3.4
  - ./configure
  - Make
  - make install
  - make install-man

It will be /usr/local/sbin.

Using Tcpdump

- **Command Line Options**
- **-n**
  - By default, tcpdump performs a DNS query to look up the hostname associated with an IP address and uses the hostname in the output.
  - to disable hostname lookups
- -s saplen
  - to grab the protocol headers, but it is not the entire packet. The snaplen option allows you to set the number of bytes tcpdump will grab from the packet
- -x
  - instructs tcpdump to print the packet contents, which it does in hexadecimal notation
- -v
  - print more information than usual about the protocols present
- -vv
  - it will print even more detailed information
- -q
  - to print less information on each line
- **-i interface**
  - If more than one interface, specify which one tcpdump should listen
- **-e**
  - include the Ethernet (or other layer 2) header information in the output
- -l
  - output to be line buffered
- **-w file**
  - to store packet data in a binary format
- **–r**
  - **read**

- The -x option
  - instructs tcpdump to print the packet contents, which it does in hexadecimal notation:

13:11:44.459933 client.example.com.48630 > server.example.com...

4510 0028 7b8e 4000 fc06 dcc4 1265 0192

0a12 0064 bdf6 0017 b6e8 5b3c 2fdc c055

5010 210c a7f7 0000 0000 0000 0000

Examples of Debugging with Tcpdump

- **Packet Flooding**
  - if you know the flooding is directed at a particular host, use a filter to view traffic destined for that host:
    - # tcpdump -n dst victim.example.com
  - #tcpdump host client.example.com

18:06:11.162372 client.example.com.45600 > dns.example.com.doma…

Maintaining Tcpdump

- no maintenance
- If upgrade the program on occasion, but it does not change very often

BASIC TCP/IP TOOLS

Basic TCP/IP Tools

- the ping program
  - performs basic tests for network reachability to a host
- the telnet and Netcat programs
  - to test application level protocol problems
- traceroute and MTR
  - determine network routing paths
- Netstat
  - prints information about network connections to a workstation.

Ping

- It can tell you if a machine is alive on the network
- it can print statistics on the network conditions from your machine to another
- The ping program
  - sends an Internet Control Message Protocol (ICMP) echo message to a remote host
  - When a networked device receives the echo message, it responds with an ICMP echo reply

Examples

- %ping workstation.example.com

workstation.example.com is alive

- % ping client.example.com

no answer from client.example.com

- ping -s server1.example.com

PING server1.example.com: 56 data bytes

64 bytes from server1.example.com (192.0.2.3): icmp_seq=0. time=14. ms

64 bytes from server1.example.com (192.0.2.3): icmp_seq=1. time=14. ms

.....

Telnet

- it is used to login to remote hosts
- but it can also be a valuable tool for network administration.
- Many networking protocols can be used in telnet
  - HTTP (for loading Web pages)
  - IMAP (for retrieving email)
  - SMTP (for sending email)
- % telnet www.example.com 80

Trying 192.0.34.166...

Connected to www.example.com (192.0.34.166)

……

Netcat

- send data to and from text-based TCP protocols like telnet
- additional features:
  - It can send and receive binary data.
  - It can handle UDP protocols.
  - It can be set up as a listener for incoming connections.
  - Data can be piped to or from a file.

Installing Netcat

- sometimes comes installed with operating systems
- /usr/bin/nc.
- http://www.atstake.com/research/tools/network_utilities/
- mkdir nc
- mv nc110.tgz nc
- cd nc
- gunzip -c nc110.tgz | tar xvf –
- make linux

When the build is complete, there will be a file called nc, which is the Netcat program.

Using Netcat

- **nc -h**
- nc [-options] hostname port[s] [ports] ...

- nc www.example.com 80
  - to test a Web server
- nc -o /var/tmp/hexout www.example.com 80
- nc -l -p 80
  - l-listen mode p-port

Traceroute

- prints a list of the routers an IP packet travels through on its way to a particular destination
- If trouble due to misrouted packets, traceroute will help identify the problem
- **How Traceroute Works**
- traceroute server.example.com

traceroute to SERVER.EXAMPLE.COM (192.0.2.50), 30 hops max, 40...
1 ROUTER-1.EXAMPLE.COM (192.0.2.1) 0.379 ms 0.273 ms 0.316 ms
2 ROUTER-2.EXAMPLE.COM (192.0.2.2) 0.335 ms 0.365 ms 0.320 ms
3 SERVER.EXAMPLE.COM (192.0.2.50) 69.641 ms 38.169 ms 39.9...

Installing Traceroute

- Most modern versions of Unix come with traceroute installed by default.
- /usr/sbin/traceroute in a root account

If not, download it

- **./configure**
- **make**
- Traceroute needs to be run with root privileges.
- **chown root /usr/local/bin/traceroute**
- **chmod u+s /usr/local/bin/traceroute**

Using Traceroute

- Traceroute will sometimes print special characters designating that a particular kind of unexpected response was received.

Character	Meaning
*	No response received
!H	Host unreachable
!N	Network unreachable
!P	Protocol unreachable
!S	Source route failed
!F	Fragmentation needed
!X	Administratively unreachable
!number	Other ICMP unreachable
!	Response TTL < 1

MTR

- Matt's traceroute (MTR) is a newer version of traceroute that combines the functionality of traceroute and ping in a single interactive session.
- **Installing MTR**
  - http://www.bitwizard.nl/mtr/
  - gunzip -c mtr-0.53.tar.gz | tar xvf –
  - cd mtr-0.53
  - ./configure
  - Make

  When it is complete, there should be an executable file named mtr in the directory, indicating that the build was successful

Using MTR

- MTR requires root privileges to run correctly.
- capture of a text-based interactive session
  - mtr -t www.example.com

Matt's traceroute [v0.52]

workstation1.mit.edu Tue Feb 25 18:41:58 2003

Keys: D - Display mode R - Restart s Packets Pings

Hostname	%Loss	Rcv	Snt	Last	Best	Avg	Worst
1. ROUTER1.MIT.EDU	0%	5	5	0	0	0	0
2. EXTERNAL-RTR-2-BACKBONE.MIT.EDU	0%	5	5	0	0	0	0
.....							
17. www.example.com	0%	4	4	83	83	83	83

Commonly Used MTR Options

Option	Meaning
-n	Do not perform DNS lookups
-g	Force use of the graphical interface
-t	Force use of the terminal interface
-h	Print help

NETSTAT

- network reporting tool for Unix workstations
- The default behavior is to print information about active network connections on a workstation
- useful for examining problems on servers and client machines alike
- **netstat**

TCP: IPv4

Local Address Remote Address Swind Send-Q Rwind Recv-Q State

- to view the routing table. Using the -r option:

netstat -r

Routing Table: IPv4

Destination	Gateway	Flags	Ref	Use	Interface
192.0.2.0	workstation	U	1	15116	eri0

Custom Tools

- two such scripting languages:
  - the Bourne shell
    - is present on every standard Unix system
  - the Perl language
    - is now installed on most modern systems.

Basics of Scripting

- programming languages
  - compiled languages
    - Most of the programs you run on a Unix system are compiled programs. You can check with the file command:
      - file /usr/bin/date

      /usr/bin/date: ELF 32-bit MSB executable SPARC Version 1, ...

      The text "ELF" and "SPARC" are tipoffs that the program is compiled
  - interpreted languages
    - file test.pl

    test.pl: executable /usr/bin/perl script

DIFFERENCES

- Compiled languages
  - to run faster
  - produce a binary file that can be run on only machines of the same processor architecture and operating system
- interpreted languages
  - Slower
  - interpreted file can run on any machine

Running a Script

- echo "Hello world"
- Store it in “testscript” file
- % sh testscript

Hello world

- **Local and Environment Variables**
  - % env

PWD=/var/tmp/
XUSERFILESEARCHPATH=/usr/athena/lib/X11/app-defaults/%N
PAGER=less
VERBOSELOGIN=1 ...

The Bourne Shell

- **Basics**
  - Bourne shell script is a list of commands
  - hostname
    date
    uptime
    Who

  hostname; date; uptime; who
- **Using Variables**

```
a="This is a test"
echo $a
```

- **Local and Environment Variables**
  - echo $EDITOR
  - EDITOR=vi
- **Conditionals**
  - a="green"
    if ["$a" = "red"]; then
        echo "Found red"
    elif ["$a" = "green"]; then
        echo "Found green"
    else
        echo "Found a color other than red or green"
    fi

- % ./scanhost myserver.example.com

```
if [ "$1" = "" ]; then
echo "You must supply a hostname"
exit 1
fi
host="$1"
echo "Scanning $host"
```

Loops

```
servers="mail1.example.com mail2.example.com time.example.com"
for i in $servers; do
echo $i
done
```

```
while [ "$1" != "" ]; do
....
done
```

Working with Input and Output

- echo "Starting script at 'date'" > /var/tmp/log
- echo "Ending script at 'date'" >> /var/tmp/log
- grep $user < /etc/passwd

Functions

```
myfunction ()
{ return 1 }
myfunction
echo $?

./testscript.sh
Result: 1
```

- trap "echo Interrupted; exit 1“
- $$ - process id
- # - comment

Reference

- Open Source Network Administration by James Kretchmar, Prentice Hall