Network Analyzer

- Network troubleshooting
- Monitoring bandwidth usage
- Defend against the security threats
- Programming troubleshooting
- Learn/Examine Network protocol

Tools

- Tcpdump
- WireShark
- Snoop
- nmap
- Snort
- Eavesdrop
- SpyNet
- CommView
- Ettercap

TCPDUMP

- Refer to book “Open Source Network Administration”
  - Online sample chapter: http://www.phptr.com/articles/article.asp?p=170902&seqNum=4
- Some tools are not based directly on the data being transmitted on a network, but information related to that data.
  - For example, network bandwidth values
  - System logs on network equipment
- Sometimes needs to examine the packets themselves.
  - Diagnose some particularly tricky network problems
- Widely used open source tool for directly analyzing packets: tcpdump
  - http://www.tcpdump.org/

Caution

- Before you use tcpdump or other analyzer:
  - Will be able to see some private data
  - Consult/research Legal implication first
  - Respect the privacy of other users

What Tcpendump can do for you

- View the entire data portion of an Ethernet frame or other link layer protocol
  - An IP packet
  - An ARP packet
  - Or any protocol at a higher layer than Ethernet.
- Very useful
  - Tcpdump is to a network administrator like a microscope to a biologist.
  - Give a very clear picture of a specific part of your network.
  - Can be used when the problem is simply that something is not working properly.

What tcpdump can do for you?

- Case 1: Web browser cannot load pages from a server - it hangs.
  - Problem with client? Server? Or between?
  - Run tcpdump while loading
    - Watch every stage to see the following
      - DNS query
      - HTTP request to server
      - Server respond
- Case 2: help debug denial of service attacks.
  - Tcp show the source address, destination address, type of traffic, etc.
  - Check the packet contents to learn more about the nature of the attack.
Limitations of Tcpdump

- Limited by network hardware
  - For example
    - Ethernet card will discard packets with an invalid checksum.
    - Tcpdump is not helpful for detecting this kind of broken packet on your network – need specialized hardware.
  - Tcpdump is showing you only what the data is, not what it ought to be – has no ability to report “IP address is forged in the packet”

Installing Tcpdump

- Already installed?
  - /usr/sbin/tcpdump
  - /usr/local/bin/tcpdump
- Download software from http://www.tcpdump.org/

Running as root

- Enable promiscuous mode
  - How?
  - Capture packets that are not addressed to the interface itself
  - Possible degraded performance
- Command line options
  - -n:
    - By default tcpdump performs DNS query to lookup hostname associated with an IP address and uses the hostname in the output. Look nicer, cause performance problem.
    - Use -n to disable it.

More command line options

- -s snaplen
  - Capture the first 68 bytes by default, enough to grab the header, not the entire packet.
  - See more data by setting snaplen to be long.
    - For ethernet, how long we can set snaplen?
- -x
  - Print the packet contents in hexadecimal notation.
- -v and -vv
  - Print more info about protocols
- -q
  - Print less info
- -i interface
  - Which one to listen on

More command line options

- -e
  - Include Ethernet header
- -l
  - Force tcpdump output to be line buffered.
  - #tcpdump -l | tee tcpdump.out
- -w file and -r
  - Store the data in binary format and then play back as it were being read from the wire using -r

Filters

- Following the command line options is the expression to dictate exactly which packets should be captured and which should be ignored.
- Primitive
  - src, dst
    - # tcpdump src client.example.com and dst server.example.com
  - Only those packet from “src” and to “dst”
  - host
    - # tcpdump host client.example.com
  - Can be combined with and, or and not with parentheses.
    - # tcpdump “host client and not (port telnet or port domain)”
Some Primitives

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>src addr</td>
<td>Source IP address matches address</td>
</tr>
<tr>
<td>dst addr</td>
<td>Destination IP address matches address</td>
</tr>
<tr>
<td>host addr</td>
<td>Source or destination IP address matches address</td>
</tr>
<tr>
<td>ether &lt;src/dst/host&gt; addr</td>
<td>Ethernet address matches address</td>
</tr>
<tr>
<td>[src/dst] net net</td>
<td>IP address is on network net</td>
</tr>
<tr>
<td>net net</td>
<td>Source or destination IP address is on network net</td>
</tr>
<tr>
<td>net net mask mask</td>
<td>As above but network range defined by mask</td>
</tr>
<tr>
<td>[src/dst] port port</td>
<td>Port is port</td>
</tr>
<tr>
<td>less octets</td>
<td>Packet size is less than or equal to octets</td>
</tr>
<tr>
<td>greater octets</td>
<td>Packet size is greater than or equal to octets</td>
</tr>
<tr>
<td>icmp</td>
<td>Packet is an ICMP packet</td>
</tr>
<tr>
<td>tcp</td>
<td>Packet is a TCP packet</td>
</tr>
<tr>
<td>udp</td>
<td>Packet is a UDP packet</td>
</tr>
<tr>
<td>arp</td>
<td>Packet is an ARP packet</td>
</tr>
<tr>
<td>broadcast</td>
<td>Packet is addresses to a broadcast address</td>
</tr>
</tbody>
</table>

Examples

- Use option and primitive, put together a number of useful tcpdump command line.
  - Display quick info on all traffic to/from a host
    - `#tcpdump -g host broken.example.com`
  - View entire packet for all bootpc traffic
    - `# tcpdump -nx 1500 bootpc`
  - To gather ssh connections and leave tcpdump running for a long time to client.example.com
    - `# tcpdump -nx 1500 -w tcpdump.data port 22 and host client`
Possible ways

- Connect the host in question and your monitoring host to a true repeater.

![Diagram of network setup](image)

Possible ways

- Configure network hardware to forward the packets you are interested in to a port you can monitor them from.
  - Not all hardware support.
  - Cisco switch is capable of doing so.

Debugging with Tcpdump

- Packet flooding

  ```
  Linux# tcpdump -n
  17:36:16.265220 10.255.255.27.1221 > 10.18.0.100.9995: udp 1168 (DF)
  17:36:16.269171 10.255.255.27.1221 > 10.18.0.100.9995: udp 1168 (DF)
  17:36:16.273130 10.255.255.23.1221 > 10.18.0.100.9995: udp 1168 (DF)
  17:36:16.285228 10.255.255.27.1221 > 10.18.0.100.9995: udp 1168 (DF)
  17:36:16.302173 10.255.255.27.1221 > 10.18.0.100.9995: udp 1168 (DF)
  17:36:16.319372 10.255.255.27.1221 > 10.18.0.100.9995: udp 1168 (DF)
  17:36:16.334600 10.7.15.65.7000 > 10.18.1.140.7001: rx ack (66) (DF)
  17:36:16.334975 10.7.15.65.7000 > 10.18.1.140.7001: rx data (36) (DF)
  17:36:16.336606 10.255.255.27.1221 > 10.18.0.100.9995: udp 1168 (DF)
  17:36:16.336623 10.7.1.70.7000 > 10.18.1.140.7001: rx ack (66) (DF)
  17:36:16.336939 10.7.1.70.7000 > 10.18.1.140.7001: rx data (36) (DF)
  17:36:16.352253 10.255.255.27.1221 > 10.18.0.100.9995: udp 1168 (DF)
  17:36:16.356199 10.255.255.27.1221 > 10.18.0.100.9995: udp 1168 (DF)
  17:36:16.396921 10.255.255.27.1221 > 10.18.0.100.9995: udp 1168 (DF)
  ```

Debugging with Tcpdump

- Webbrowser hangs

  - Step 1. Start tcpdump to monitor port 80
    ```
    #tcpdump host client.example.com and port 80
    ```
    - No traffic, so web server is not a problem
  
  - Step 2. Start tcpdump for all
    ```
    #tcpdump host client.example.com
    ```
    - Only DNS request, no response:

      ```
      Linux# tcpdump -xls 1500 host client | ./tcpdump-data-filter.pl
      18:14:12.842409 brokenclient.example.com.55313 > dns.example.co...
      4250 0104 010f 058b 4000 ff11 9d80 0a12 0064      E..H..@........d
      0a05 061e d811 0035 0034 8a44 e4ca 0010      .......5.4.D....
      0001 0000 0000 0001 0377 7777 0765 7861      .........www.exa
      6d70 0363 6f6d 0000 0f00 0100 0029      mple.com.......)n
      0800 0000 8000 0000
      ```

Graphs

- One of Network Administrator’s important tasks: Monitor bandwidth, network usage
  - Current status
  - Trends

Pictures


Amount of traffic sent INTO from an interface

Amount of traffic sent OUT from an interface
MRTG

- MRTG stands for Multi Router Traffic Grapher
  - Produce web pages that display graphs of bandwidth use on network links on daily, weekly, monthly and yearly scales.
  - Rely on SNMP
  - MRTG is based on Perl and C and works under UNIX and Windows NT.
  - Free software released under GNU General Public License.

How MRTG works

- Send SNMP requests regularly
  - Like every 5 minutes
- Store the response in a specialized data format
- Summarize the data and create PNG files (Portable Network Graphics)
  - The PNG files can be included in the web page or other applications
- Create html files

What MRTG can help you Do

- View the normal traffic pattern
  - Studying the trend
  - Plan capacity needs for future
- Quickly determine abnormal traffic load
  - Use graphs of history
    - Sudden change might count for an operational problem
    - A denial-of-service attack

What MRTG can help you do

- MRTG is most often used to collect data from router interfaces
- Can also collect traffic data from switches or servers
- It can be configured to collect any statistical data that a device makes available via SNMP.

Installing MRTG

- Available at http://www.mrtg.org
- It requires:
  - Perl 5.005 or greater
  - The GD library
  - The PNG library
  - The zlib library
- MRTG comes with its own SNMP implementation.
- Follow the doc to

Configuring MRTG

- Configuration File
  - Read only by the user who is running the process
    - Should not be public readable
    - Include the community names for your devices
  - Specify where to put data
    - Under a Web Server
    - Publicly readable
Configuring MRTG

- Use utility cfgmaker with lots of options to create the configuration file
  - Global options
  - SNMP options
  - Monitored device and community name
  - Use IP as directory name in data area.
- Example: Creates a config file
  - for router.place.xyz: the router has the community name public.
  - Interfaces get identified by their IP number.
  - The config file gets redirected to mrtg.conf.

# cfgmaker --global "WorkDir: /var/www/htdocs/mrtg" \
--global "Options[_]: growright,bits" \
--ifref=ip "public@router.place.xyz" > mrtg.cfg

Generating Data

- Generating initial data
  - Run mrtg command with the configuration file on the command line
  - View view the data http://.../mrtg
- Setting up regular data gathering
  - Crontab to run mrtg utility (recommended)
  - Run mrtg as a daemon with options
    - Require change to the configuration file to ass "RunAsDaemon Yes"
- Use indexmaker program to organize the PNG file and HTML file

More about MRTG

- Do not trust it blindly.
  - Does data make sense?
    - Bugs?
    - Counter overflow problem?
    - Missing data?
      - Could not retrieve or store data for a short period time
      - Data value will be flat because it uses the data from previous interval
- Maintaining MRTG
  - Requires some maintenance
    - Every time moving a network or router interface, make sure the change is reflected in the MRTG configuration

NETFLOW

- NETFLOW Available on some routers, such as Cisco, Juniper.
- Collecting data
  - Source and destination IP addresses
  - Source and destination protocol port numbers
  - Number of packets transmitted
  - Number of bytes transmitted,
  - ...
- View the data on the router
- Forward the data to another host

What NetFlow can help you do

- Image your network is hit by a denial of service attack.
  - You noticed degraded network connectivity
  - MRTG show dramatic rise in traffic levels
  - Interface counter on router indicate a very high rate of traffic
- Turning on NetFlow on the router
  - Examine the traffic in realtime
  - Notice a large number of connections from a single host, all the sequentially increasing IP addresses inside your network - scanning?
  - Block the traffic at the border router?

How NetFlow works

- Based on the idea of a flow of network traffic.
  - Flow is one full network conversation from the start to the end.
  - Every flow has a unique set of properties:
    - Source IP address
    - Destination IP address
    - Source port
    - Destination port
    - IP protocol number
    - Type of service field
    - Input Interface
  - For example,
    - one tcp connection to a web page is a flow
    - Scanning destination IP addresses will generate many flows.
**NetFlow**

- Flow is unidirectional.
  - Routers report flows for traffic that enters an interface, not traffic that leaves an interface.
- How to determine what constitutes one sessions’ worth of traffic and when a particular flow ends?
  - For TCP is relatively easy.
    - The session begins with a TCP SYN, and ends with a TCP FIN.
  - For UDP, ICMP, etc, it is more difficult. The router must use heuristics to decide when the flow is complete.

**NetFlow**

- Router has limited memory for holding data.
- It removes flows from cache occasionally.
- Router will expire flow if:
  - The end of a tcp connection is found.
  - No traffic has been present in the flow for 15 seconds.
  - The flow has been running for over 30 minutes.
  - The table of flows in the router is filled.
- One real flow could become multiple flows due to router expiration policy.

**Exporting NetFlow Data**

- Routers can export NetFlow data.
  - To a server, called flow collector, then to be processed by administrator.
  - As UDP packets to an IP address:
    - Data may be lost.
    - Only one IP address.
    - Flow collector has to forward to other flow collector if you wish.
    - Router address is not in the packets after forwarded again unless some other mechanism is used.

**NetFlow Versions**

- Version 1, each flow contains:
  - Source and destination IP address.
  - Source and destination port address.
  - Next hop router address.
  - Input and output interface number.
  - Number of packets and bytes sent.
  - sysUpTime when flow began and ended.
  - IP protocol number and type of service.
  - Logical OR of all TCP flags seen.

- Version 5
  - Include AS numbers.
  - Sequence numbers, which allows checking for lost packets.

- Version 7

- Version 8
  - Router-based NetFlow aggregation.
  - Reduce the data send to collectors.

- Version 9
  - Template-based scheme for reporting flows.

**Installing Flow-tools**

- OSU Flow-tools is available form:
  - [http://www.net.ohio-state.edu/software](http://www.net.ohio-state.edu/software)
  - Download.
  - Extract.
  - Build.
  - Install.
- Configuring NetFlow on the router:
  - Enable NetFlow on some interfaces.
  - Configure router to export flows.
Using Flow-tools

Capturing Flows
- Flow-capture: receive and store
- Flow-receive: receive and send to standard output
- Source/destination and port number can be used as criteria to receive the NetFlow info.
  - For example
    ```bash
    #flow-receive 0/0/9995 | flow-print
    0/0/9995 has the format of localip/remoteip/port
    ```
- Note: this is related to communication of sending and collecting flows, not the addresses contained within the flows.

Using Flow-tools

Flow-capture
- By default:
  - Create a new file every 15 minutes
  - Create a nested directory structures so that each year, month and day of files has a separate directory
  - Never remote files
  - Compress files
- Default settings can be changed from command line
  - Set up a maximum size that data files may be use in total
  - Change the directory hierarchical structure
  - File rotation rate
  - Compression rate
  - Remote client

Using Flow-tools

Viewing the flow data
- Flow-print
  - ASCII
  - Predefined reporting format available
- Flow-report
  - Perform some statistical analysis
  - Produce the data in a way to be able to feed other tool
  - Require a configuration file
- Flow-stat
  - Does not require a configuration file, has predefined formats
  - Similar to flow-report
- Flow-dscan
  - Attempts to detect unusual network traffic like port scanning and host scanning.

Using Flow-tools

Manipulating Flow Data
- Flow-cat and flow-merge
- Flow-split
- Flow-expire
- Flow-header
- Flow-fanout

Other Netflow tools

Netflow Commercial software packages

Video time
- Scrutinizer
  - http://www.youtube.com/watch?v=YZI_dOAwys0
  - http://www.youtube.com/watch?v=Kry6tpTjBYANR=1
  - http://www.youtube.com/watch?v=ZnT1MFcDTM&feature=related

Service Monitoring

Network administrators has to deal with something that spontaneously stops working.
- Network link fails
- Web server refuses connections
- Switch stop passing traffic
- ...

Network administrators need to be notified of failures when they occur. How?
Service Monitoring

- Monitoring or polling software
  - Send out probes at regular intervals to test
    - Network connectivity
      - Ping switch, router, hosts
    - Service functionality
      - Attempt to retrieve web page
      - Make snmp request
      - Perform other service level testing ...
  - Has to be intelligent to report the root problem.
    - For example
      Polling server $\rightarrow$ switch A $\rightarrow$ switch B $\rightarrow$ many many hosts
    - If switch B fails, should Network Administrator receive message regarding each hosts connected to switch B?
    - Should not degrade performance too much - polling intervals

What Service Monitoring Can Help you do?

- Give you early warning of failures
  - Proactive management, not fire fighting
  - Fix a problem before it becomes an even larger problem
    - For example;
      - If the redundancy part failed without being noticed, what will happen when the primary part fails?
- Help you determine where is your network a problem resides.
  - Depends on user report and waits for user to complain is too late sometime.
  - It is always in your best interest to know about problems before either your customers or your superiors do.

Open source software

- Sysmon
  - Available at http://www.sysmon.org
  - Simple program, easy to configure and get running
  - Does not have many advanced features
  - Works for a small to medium-sized network
- Nagios
  - Available at http://www.nagios.org/
  - Is a better tool for large networks
  - Complicated program
  - Includes advanced functionalities.

Installing and Using Sysmon

- Installing Sysmon
  - Where to place the server?
    - Use a stable server
    - In a stable place, near the network core - you don’t put sysmon on a tend-to-fail place.
  - How to install Sysmon
    - Download, unpack and build the software
  - Using Sysmon
    - Create a configuration file
    - Control daemon

Configuring Sysmon

- Configuration is made up a list of objects to be monitored.
  - IP address of the device to be tested
  - The kind of test that should be performed
  - Any objects the device depends on

For example:

Object router1 {
  ip “192.0.2.5”;
  type ping;
  desc “Router1”;
  dep “server”;
  contact admin@example.com
};

Setting the IP address

- Really contain either hostname or IP
- If ipaddress, then sysmon is not depend on DNS
- If many pieces to be monitored, using host name is easier

Setting the Test Type

- Pong standard ping test
- Pop3 working POP3 server username,password
- Tcp generic listening TCP port port
- Udp generic listening UDP port port
- Radius working Radius server username,password,secret
- Nntp listening news server
- Smtp listening mail server
- Dnmp listening DMAP server
- X5000 listening x5000 directory server
- Ww listening web server url, urifield
**Sysmon**

- **Example**
  
  ```
  Obejct web-server {
  ip www.example.com;
  type www;
  desc "Main Web Server";
  dep "Router1";
  url http://www.example.com/;
  urltext "<TITLE>";
  contact admin@example.com;
  }
  ```

**Sysmon**

- **Specifying Dependencies**
  
  ```
  Object server1 {
  ip "server1.example.com";
  type ping;
  desc "Server 1";
  dep "Router1-servernet";
  dep "Router2-servernet";
  contact admin@example.com;
  }
  ```

**Sysmon**

- **Using the spawn options**
  
  - Email is not always the best way to notify an administrator of a critical problem.
  - Email is not guaranteed to be a timely service.
  - What if mail system itself is unavailable?
  - Send a direct message to a page or cell phone is better.
  - Use spawn allows to hook a program to execute with arguments:
    - User the replacement variables to create as detailed as you would like:
      - `%m` local host name
      - `%s` service
      - `%p` port number (numeric)
      - `%T` current Time hh:mm:ss

**Sysmon**

- **Global options**
  
  - The status file
    - Sysmon daemon periodically writes a file with the status of services that is monitoring.
    - Configurable parameters:
      - File type: html or text
        - Change the color scheme for html format
      - Write time interval
      - View both up and down services
    - Configure the header of mail to include:
      - From
      - Subject

**Sysmon**

- **More global options**
  
  - Test queuing options - control how Sysmon processes service tests and notifications.
    - How many tests a service must fail before a notification message is sent.
    - Numfailures
    - Queuetime
    - Maxqueued
    - How many test Sysmon can run simultaneously
    - How frequently should repeated messages been sent
    - pageinterval

**Sysmon**

- **Using variables**
  
  - If some value, like email address, is used many times, define a variable for it and reference it using $var

**Sysmon**

- **Using Includes**
  
  - Break the configuration down info smaller files
    - One for web servers
    - One for ping tests
    - ...

**Sysmon**

- **Maintaining Sysmon**
  
  - Keep configuration accurate - toughest
  - Otherwise, it became useless for new changes or "cried wolf" for old failures
Nagios

- Open source: http://www.nagios.org
- Major advantages
  - Escalation
    - Email when failed first time
    - Page if happened again
  - Configuration templates
    - Only configure the difference
  - Monitoring time periods
    - During work hours
    - Scheduled downtime - do you want to be paged in the middle of the night when somebody else is upgrading a piece of equipment?
  - Modular test plugins
    - Each plugin is simply an external program that tests a service
    - It is easy to write your own tests to complement the suite of tests

Nagios

- Passive tests
  - Some information cannot be sent to the monitoring server by means of the server requesting of the data
    - SNMP traps
  - Host and contact groups
    - Easier to change configuration for a large, similar set of devices all at once.
  - Flap detection
    - Test repeated fails and succeeds - flapping
    - Automatically disable notification until the flapping has stopped.
  - Optional dependencies

Nagios

- Downside
  - Much more complicated program than Sysmon
  - Take a significant amount of time to install and configure
    - Sysmon can be setup in an afternoon
    - Nagios may take several days or longer

- Quick start guide:
  - http://nagios.sourceforge.net/docs/3.0/quicks tart-fedora.html

Summary

- Package Analyzer
  - TCPDUMP
  - Wireshark
- Bandwidth Grapher
  - MRTG
- Router Traffic monitoring
  - NETFLOW
- Service Monitoring
  - SYSMON
  - Nagios