Network Administration

Preetam Patil

KReSIT, IIT Bombay

mailto:yogi@it.iitb.ac.in

Network Administration

GOAL: Keeping the network running properly, and configuring and managing services that are provided over the network!

- There are many services that we use regularly ...
- and there are some which work in the background (e.g. DNS) enabling other services to run smoothly!
- We'll take a look at some of them:
  - DNS
  - E-Mail
  - Firewalls and Proxies
  - FTP
  - some misc utilities

Domain-name System

Address resolution - Earlier

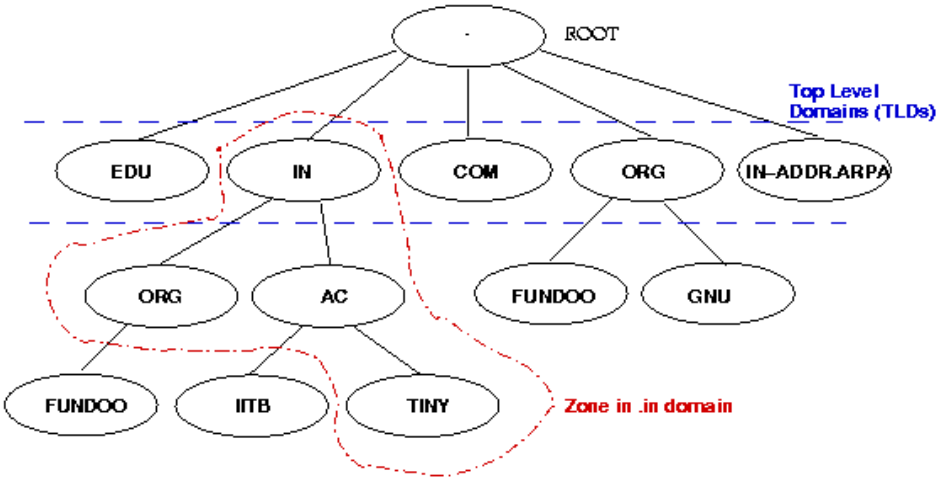
- The name-to-address mapping of all hosts known were used to be stored in all hosts, in `/etc/hosts` file
- There used to be periodic updates to this mapping file
- This worked well till the size of Internet was small, but couldn't be continued because
  - new hosts and names were being added frequently, so keeping the file updated was problematic
  - name space collision- two hosts could possibly choose the same name independently, causing collision
  - administrative authority- different networks were under different administrative controls, and there was no reason why you needed to update global database for local hostname changes

Solution: building distributed hierarchical database

- Called Domain Name System which is a tree of domains! “DNS is a set of protocols for distributed database”
- The network is broken into a hierarchy of domains
- The namespace is organized as a tree according to organizational or administrative boundries
- Each node, called a domain, is given a label, and the name of the domain is the concatenation of all the labels of the domains from the root to the current domain, listed from right to left, separated by dots
- A label need only be unique within its domain

Namespace Organization

- Moreover, the namespace is partitioned into several areas called zones, each starting at a domain and extending down to the leaf domains or to domains where other zones start zones usually represent administrative boundries



Tree of Domains

What is a domain?

- A domain is a registry which may contain:
  - definitions of subdomains and information about how to reach one
  - address of contact person for the domain
  - name-to-address mappings (or the reverse way)
  - information about how to route mails for the domain
  - information about the well-known services provided by the domain

Resolvers

- A resolver is a client of the DNS
- Resolvers are used by networking applications to query the DNS
- Resolvers direct the queries at name servers that contain parts of the distributed database, using the DNS protocols
- The resolver libraries are located in the application layer of the networking software of each TCP/IP capable machine

BIND (Berkeley Internet Name Domain)

- Consists of a DNS Name-server (called named) and resolver libraries
- BIND deals primarily with zones
- Types of servers
  - Caching only server:
    - a Caching Only Server is a server that is not authoritative for any zone
    - this server queries and asks other servers, who have authority for any zone
    - all servers keep data in their cache until the data expires, based on TTL("Time-to-live") field which is maintained for all resource records

BIND (Berkeley Internet Name Domain)

- Types of servers (contd...)
  - Slave server:
    - a server that always forwards queries it cannot satisfy from its cache, to a fixed list of forwarding servers instead of interacting with the authoritative servers
  - Authoritative servers:
    - the servers which are authorized to answer qeuries for any entries in their zones
    - They are further classified as Primary and Secondary
    - Primary servers ae the servers where the records for the zone for which it’s authoritative are maintained
    - Secondary servers get the zone records from the primary server

BIND configurartion Files

- Boot File (/etc/named.conf)
  - this is the file that is read when named starts up
  - this tells the server what type of server it is, which zones it has authority over and where to get its initial data
- Resolver Configuration (/etc/resolv.conf)
  - designates the name-servers on the network that should be sent queries
  - this is the file referred by the resolver in the system
- Cache initialization file (/var/named/root.cache)
  - tells the nameserver about which are the authoritative nameserver for the root of the domain (this is the starting point for any lookup)

BIND configurartion Files (contd . . .)

- Zone data files:
  - hosts: contains all the data about machines and subdomains in this zone
  - hosts.rev : this file specifies the IN-ADDR.ARPA domain (this is a special domain for allowing address-to-name(reverse) mapping)
  - named.local : this file specifies the address-to-name mapping for the local loopback interface, known as localhost

Format of a Zone data file

- The zone data file is specified using Standard Resource Record Format
- The format specifies different types of objects, like
  - A: name-to-IP address mapping
  - MX: Mail exchanger for the domain
  - NS: authoritative Name-server for the domain
  - PTR: IP address-to-name(reverse) mapping
  - CNAME: Canonical name- providing aliases

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 14 of 82

Go Back

Full Screen

Close

Quit

DNS Utilities

- nslookup: used for looking up DNS data; provides interactive interface if not specified domain name on command line
- host: a simple utility for performing DNS lookups
- dig: a flexible tool for interrogating DNS name servers, used mainly by DNS administrators to debug DNS problems

E-Mail Services

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 16 of 82

Go Back

Full Screen

Close

Quit

E-mail Services

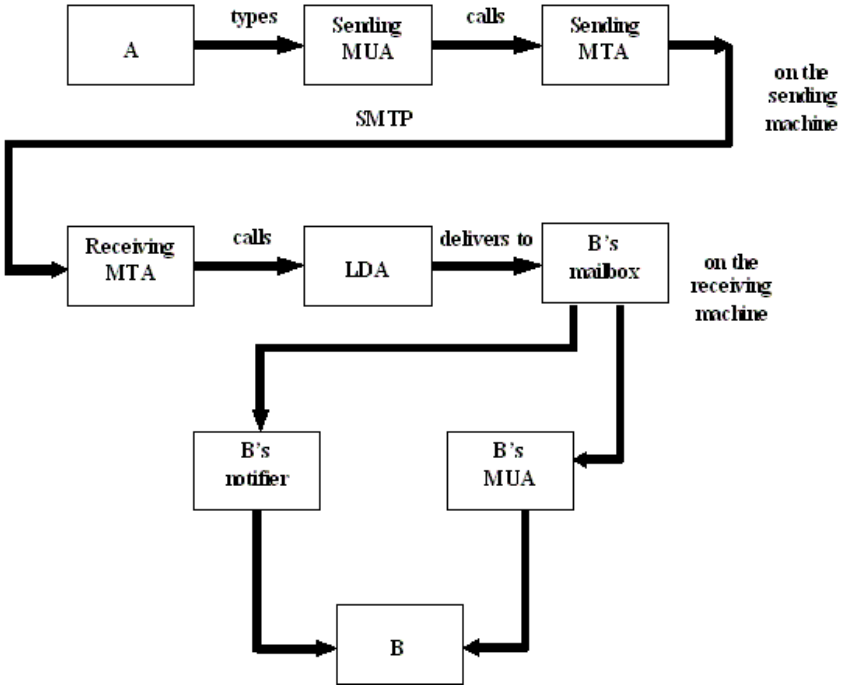
- E-mail: the most important application on Internet!
- This is because of the following features of e-mail systems:
  - the protocol is robust and reliable
  - there are a variety of ways one can access e-mail
  - it's flexible – it works with any type of network access
  - it's almost instant, but without requiring everyone to be online
- these features are the result of the many components which we'll take a look at!

How is Mail delivered?

There are four important components of email system:

- MUA - Mail Users Agent
  This is the program a user uses for sending/reading emails
- MTA - Mail Transfer Agent is used to pass mail from the sending machine to the receiving machine over the network
  There is a MTA program running on both ends
- MDA - Mail Delivery Agent on the receiving machine receives the mail from its MTA
- SMTP - Simple Mail Transfer Program Protocol is used by the MTAs on both machines to pass mail between them
  SMTP usually runs on port 25

Mail between full-time Internet machines



Home Page
Title Page
Page 18 of 82
Go Back
Full Screen
Close
Quit

UUCP

- UUCP stands for Unix-to-Unix copy
- UUCP can be used for mail transport
- Salient Features of UUCP
  - can run on different types of networks
  - efficient on resources
  - allows local mail accounts, global addresses
  - creating UUCP domains requires minimum configuration

Mail User Agents (MUA)

- Many choices of MUAs, also referred to as mail-readers available, as discussed in previous sessions
- text-based mail-readers: mutt, pine, mh, elm etc.
- GUI-based mail-readers: evolution, kmail, mozilla mail client, etc.
- most of them also include an editor, and address-book functionality
- added functionalities available: threaded sorting, mail filtering, role-playing, text searching etc.

MUTT and PINE

- MUTT
  - Supports color terminal, MIME, threaded sorting mode.
  - a powerful mail client which is extremely configurable through the configuration file, typically `.muttrc`
- Pine
  - another powerful mail client which is configurable through menu-based setup as well as config file
  - one unique feature of Pine is it's support for roles: different user profiles based on a set of conditions

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 22 of 82

Go Back

Full Screen

Close

Quit

Mail Transport Agents

- Different MTAs differ in the features they offer, flexibility of configuration, and ease of usage/administration
- the important ones are Sendmail, Exim and Qmail
- since mail server configuration is complex, they all come with tools to manage the configuration
- all of these are capable handling bulk emails

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 23 of 82

Go Back

Full Screen

Close

Quit

Sendmail

- Sendmail is believed to be one of the most complex softwares ever written, the proof of which is the sendmail configuration file `sendmail.cf`
- but fortunately, there are tools to generate the configuration using predefined macros
- moreover, there are many sensible defaults which need not be tweaked unless needed

Sendmail functionalities

- accepting/making SMTP connections (over network/locally)
- local mail delivery
- mail relaying
- many user database options
- access control for relaying, receiving, delivery on per-user, per-host or per-domain basis
- support for spam-blocking from a list of spam sites which is retrieved automatically
- range of operations on mails
- support for many security enhancements
- can be combined with other mail-processing softwares for even more functionality- e.g with procmail for powerful mail filtering

Sendmail Configuration

- Sendmail Files:
  - `/etc/mail`: default configuration directory
  - `/etc/aliases` contains the aliases source listing
  - `/etc/sendmail.mc` sendmail macro configuration file
  - `/etc/mail/sendmail.cw` list of domains for which to accept mails for
  - `/etc/mail/access.txt` sendmail access configuration file
  - `/var/spool/mqueue/` contains the mail files till they are not delivered
  - `/var/spool/mail/username` contains the user's mailbox

Sendmail Configuration (contd . . .)

- Typical steps in Sendmail configuration are:
  - The `sendmail.mc` macro configuration file is the starting point for sendmail configuration
  - Utilities like `sendmailconfig` even help you to generate the `sendmail.mc` file
  - edit this file to put any additional features, if needed
  - edit the `access.txt` file to define access control rules, if you want access control
  - generate the access.db file by running `makedb` utility
- Maintaining user aliases
  - edit the aliases file `/etc/aliases`
  - run `newaliases` to generate the aliases database

Sendmail Utilities

- `mailq` prints the list of entries in mail queue
- `mailstats` displays the current mail statistics
- `makemap` builds desired databases from source file

Fetchmail

- Fetchmail is a very useful mail-retrieval and forwarding utility
- it fetches mail from remote mail-servers and forwards it to your local (client) machine's delivery system
- it can also be run in a daemon mode to repeatedly poll one or more remote mailboxes at a specified interval
- if fetches mail from remote server using POP or IMAP protocols
- can also be useful as a message transfer agent for sites which refuse SMTP transactions

How it works

- Fetchmail connects to the specified servers using the specified protocols, authenticates itself using the username/password specified, and retrieves mail
- it normally delivers mail via SMTP to port 25 on the machine it is running on(localhost)
- the mail is then delivered locally via the local system's MDA
- If no port 25 listener is available, it can also use that MDA for local delivery directly

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 30 of 82

Go Back

Full Screen

Close

Quit

Fetchmail configuration

- `/etc/fetchmail.conf` is the config file for fetchmail run in daemon mode
- `~/.fetchmailrc` is the config file for user-level fetchmail configuration
- Fetchmailconf is the GUI-based utility to create/modify these files

Mail Message Format

- A mail message consists of a Mail header and a Mail body separated by a blank line
- The header contains, amongst other things,
  - Source address of the mail
  - Destination address of the mail
  - Subject line
  - Date the mail was sent

Procmail-The email filter

- it’s a powerful mail processor with lots of features
- it is typically used as a mail filter that receives the mail when it arrives for the particular user
- it can also be used as local mail delivery agent
- its operation is defined by filter rules referred to as recipes
- there are great variety of things that can be done with these recipes!

Procmail Configuration

- Procmail recipes
  - a typical recipe can have the form:
    ```
    :0
    * $^To:.*mpls.uu.net.*
    /home/myname/mail/mpls
    ```
  - procmail goes through the recipes one-by-one from top to bottom to decide on the action to take on the mail
- create a `.procmailrc` file consisting of recipes you want:
- generally, mails have to be diverted to the `procmail` program through `.forward` mechanism: put the following line in the `.forward` file in you home-directory:
  `"| /usr/bin/procmail"`

Mail Notifiers

- Notifiers are the programs that inform the user of an incoming mail
- Notifier requires two programs:
  - biff - Allows the comsat service to be turned on and off.
  - comsat - Notifies the user of new mail.
- many other X-based mail notifiers like xbiff, gbuffy are available

Mailbox formats

- mbox - Mailbox format, puts each mailbox into a directory of files.
- BABYL - An old mail system.
- MMDF - The simplest. Older and crude.
- MH - Mailbox format, puts each mailbox into a directory of files.
- qmail

Internet Mail Access Protocol(IMAP)

- IMAP stands for Internet Message Access Protocol
- It is a method of accessing electronic mail remotely
- Key features of IMAP are:
  - allows message access and management from more than one computer
  - allows access without reliance on less efficient file access protocols.
  - provide support for “online”, “offline”, and “disconnected” access modes.
  - support for concurrent access to shared mailboxes.

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 37 of 82

Go Back

Full Screen

Close

Quit

Post Office Protocol(POP)

- typically used to allow a workstation to retrieve mail that the server is holding for it
- IMAP v/s POP
  - IMAP is an online protocol whereas POP is an off-line protocol
  - POP has a minimum use of connect time
  - POP uses lesser server resources than POP
  - IMAP provides access to your inbox from different computers
  - IMAP has Faster start-up time, as only message headings are transferred initially
  - IMAP is supposed be a functional superset of POP

Firewalls and Proxies

Home Page
Title Page
◀◀ ▶▶
◀ ▶
Page 39 of 82
Go Back
Full Screen
Close
Quit

What are Firewalls?

- According to “Firewall-HOWTO”
  - “Internet firewalls are intended to keep the flames of Internet hell out of your private LAN”
  - Or, “to keep the members of your Lan PURE and chaste by denying them access the all the evil Internet temptations
- Firewalls are used:
  - For securing the network, or
  - For securing your own system, i.e for dialup users or on insecure LANs
  - and to keep people(employees/children) in or, to keep a watch on insiders’ net access.

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 40 of 82

Go Back

Full Screen

Close

Quit

Filtering Firewalls

- Work at the network level (TCP and IP layers)
- Transparent to the users
- Data packets filtered based on their type, source/destination address,and port numbers
- Can be achieved efficiently, thus with less latency
- No scope for user identification/passwords
- The only user identity is the IP address

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 41 of 82

Go Back

Full Screen

Close

Quit

Proxy Servers

- “Proxies” on behalf of the client machine behind the firewall
- Finer control and monitoring of the network traffic that goes through (or isn’t allowed to go through!)
- Some proxies also cache data for bandwidth efficiency
- Not transparent to the users: i.e, users need to modify application settings to access services outside network boundry

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 42 of 82

Go Back

Full Screen

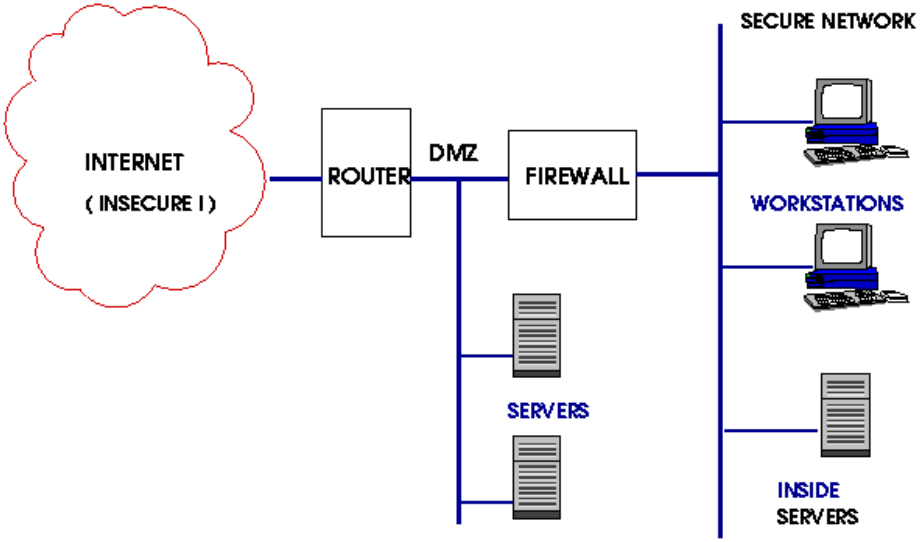
Close

Quit

Proxy Servers (contd . . .)

- Application Proxies
  - work at application layer. e.g: web(HTTP) proxy
  - user authentication through variety of means
  - can even filter “inappropriate” words or viruses
- SOCKS proxies
  - Work at Sockets level, in a manner similar to filtering proxies
  - Do not provide as much functionality as application proxies

Firewall Location



Running the Filtering Firewall

- Hardware required:
  - any standard PC, config depending on the load expected, from a 486 to P-II
  - More than one network cards, if you want to forward packets also
- Software:
  - linux kernel 2.2 and above, with firewalling features compiled in (most default kernels now a days have these!)
  - the iptables (or ipchains for 2.2.X linux kernels) package

Preparing the system

- Select/compile the kernel with firewalling features
- Configure the network interfaces
- Optional: turn on IP forwarding (if you need it)

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 46 of 82

Go Back

Full Screen

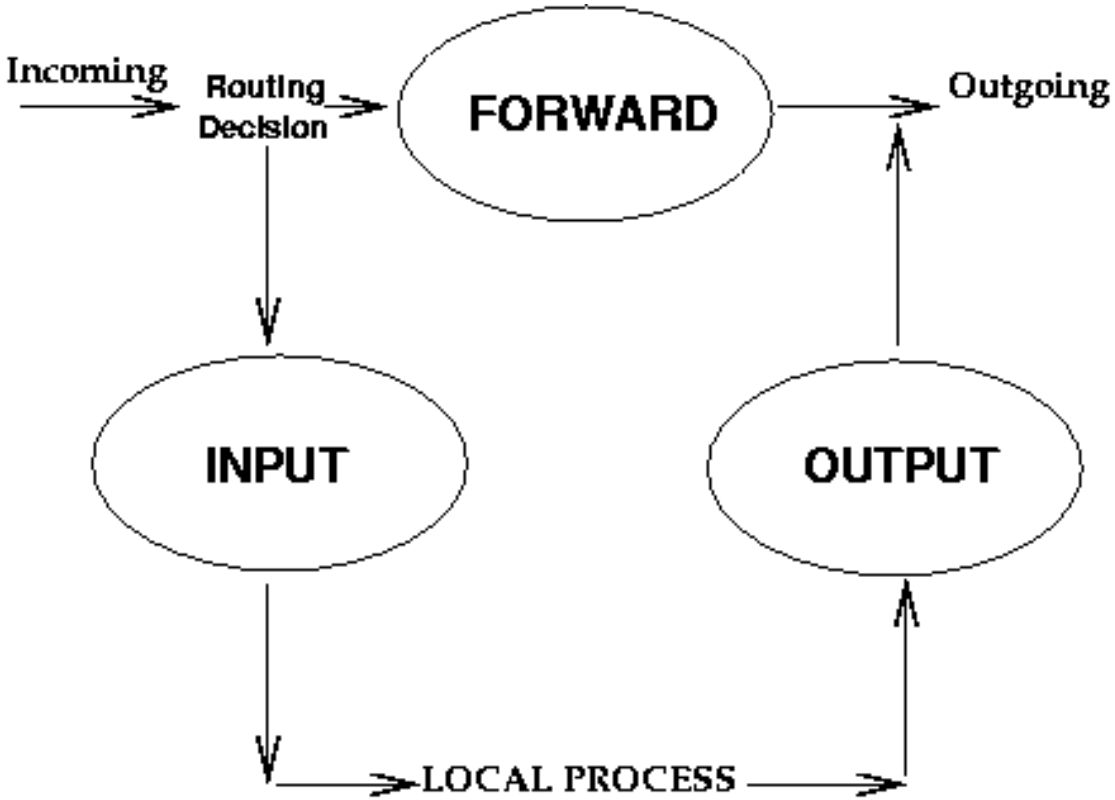
Close

Quit

IPTABLES

- IPTABLES: a combination of kernel and user-space utilities to manage packet filter from user-space
- IPTABLES consists of tables, one of them is “filter”
- Tables consist of chains, which are lists of rules
- There are three in-built chains in the “filter” table:
  - INPUT, OUTPUT, FORWARD

Chains Diagram



Operations on chains

- Creating a new chain:
  ```
  # iptables -N my_chain
  ```
- Deleting a new chain:
  ```
  # iptables -X my_chain
  ```
- Flushing a chain(deleting all rules from the chain):
  ```
  # iptables -F OUTPUT
  ```
- Listing a chain:
  ```
  # iptables -L INPUT
  ```
- Resetting the counters: `#iptables -Z FORWARD`
- Setting Policy(for in-built chains only): can be either ACCEPT or DROP
  ```
  # iptables -P FORWARD DROP
  ```

Rules

- A rule specifies a set of conditions the packet must meet, and what to do if it meets them (a ‘target’)

Operations on a single rule

- Append or Delete:

```
# iptables -A INPUT -s 192.168.0.0/16
-j ACCEPT
# iptables -D INPUT -s 192.168.0.0/16
-j ACCEPT
```

- Insert or Replace: apply to a position in the rule-list

Filtering Specification

- Specifying Source and Destination IP address
  - can specify host/domainname, address, or address/netmask

```
# iptables -A FORWARD -s 192.168.1.0/24
-j ACCEPT
```

- Specifying inversion
  - many flags can be preceded by ! to indicate inversion (NOT)

```
# iptables -A FORWARD -s ! 192.168.2.0/24
-j DROP
```

- Specifying Protocol
- protocol can be specified with the -p flag: common values are TCP, UDP and ICMP

```
# iptables -A FORWARD -d 192.168.1.0/24
-p ICMP -j ACCEPT
```

Filtering Specification (contd . . .)

- TCP Extensions: if -p TCP is specified, following extensions are available:
  - `--tcp-flags`:

```
# iptables -A FORWARD -i ppp0 -p tcp
 --tcp-flags SYN,SRT, ACK SYN -j DROP
```

  - `--sport`:
    specifies a single TCP port or a range of ports as source

```
# iptables -A INPUT -p tcp --sport 20
-j ACCEPT
```

  - `--dport` : specifies destination port, similar to sport

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 53 of 82

Go Back

Full Screen

Close

Quit

Filtering Specification (contd . . .)

- Specifying an Interface
  - `-i` flag specifies the incoming interface, `-o` specifies outgoing interface to match
    Only FORWARD chain has both input and output interface

```
# iptables -A INPUT -i ppp0 -s 192.168.0.0/16
-j DROP
```

Target Specification: specifying the action on the packet

- Target of a Rule can be:
  - one of the two simple built-in targets: DROP and ACCEPT
  - a user-defined chain: packet begins traversing rules in that user-defined chain

General tips

- Since firewalls are the exposed part of your network to the outside world,
  - run as few services as you can on the firewall system
  - avoid keeping normal user accounts on the system
  - always keep the softwares/kernel updated/patched on the firewall
  - run sanity checks on the firewall routinely
- IPtables etc are not complete firewalls! they can break your network
- traffic: if the services on your system are limited, consider application-level filtering to make your life easier.

Notes

- netfilter is a set of hooks inside the linux 2.4.x kernel's network stack which allows kernel modules to register callback functions called every time a network packet traverses one of those hooks.
- iptables is a generic table structure for the definition of rulesets.Each rule within an IP table consists out of a number of classifiers (matches) and one connected action (target).
- netfilter, iptables and the connection tracking as well as the NAT subsystems together build the whole framework.
- What should be the order of rules in the table? Order is important: action is taken based on the first rule that matches.

Proxy Servers

Why use Proxy Servers ?

- Security, crossing firewalls
- access control policies
- efficient use of resources due to caching
- junk content filtering
- accounting
- load balancing

Squid

- Squid is a high-performance proxy caching server for web clients, supporting FTP and HTTP data objects among others
- Squid consists of
  - A main server program *squid*
  - A Domain Name System lookup program *dnsserver*
  - Some optional programs for performing authentication etc

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 60 of 82

Go Back

Full Screen

Close

Quit

Squid Configuration

- Squid configuration needs to define the addresses (IP address + port) for every relevant server and gateway.
- A Squid daemon program will need to communicate with
  - Local or remote web servers
  - Other Cache servers
  - Clients (desktop browsers or gateways)
- Subsequent sections cover the important configuration options

Configuration (contd...)

- http_port
  - This tag name is used to specify the socket address where squid will listen for HTTP client requests.
  - default value is 3128
  - Usage:
    ```
    http_port port
    ```
- cache_mem
  - Specifies the ideal amount of memory used for caching
  - Default value is 8 MB
  - usage:
    ```
    cache_mem 1 GB
    ```

Proxy Access Control

- http_access
  - Allowing or denying http access based on defined access lists
  - If none of the “access” lines cause a match, the default is the opposite of the last line in the list
    e.g if the last line is deny, then the default is to allow and vice-versa
  - Examples
    *
```
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access deny all
```

Configuration (contd...)

- acl
- Usage:
  `acl aclname acltype string1 ...`
  This is used for defining an access List
- acltype: proxy_auth
  - User authentication via external process. It requires an external authentication program to check username/password combination

FTP

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 65 of 82

Go Back

Full Screen

Close

Quit

ftp: File Transfer Protocol

FTP provides means for file transfers over the network. It also has support for managing files on remote machine and user authentication. On the remote host, a server program, typically called `ftpd` needs to be running, while on the user end, any ftp client program needs to be run

The commands for ftp operations, which are supported by both ftp clients and servers, are specified by the ftp protocol, which is an Internet Standard

ftp Commands

- `binary`

  Set the file transfer type to support binary mode transfer.

- `bye`

  terminate the FTP session with the remote server and exit ftp.

- `cd`

  Change the working directory on the remote machine to *remote-directory*

- `ls [remote-directory]` Print a listing of the contents of *remote-directory* on the remote machine

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 67 of 82

Go Back

Full Screen

Close

Quit

ftp Commands(contd . . .)

- `get remote-file [local-file]` Retrieve the *remote-file* and store it on the *local machine*
- `put local-file [remote-file]` Store a *local file* on the remote machine.
- `mget remote-files` Expand the *remote-files* on the remote machine and do a get for each file name thus produced.
- `mput local-files` Similar to mget, for uploading files
- Aborting a file transfer: To abort a file transfer, use the terminal interrupt key (usually Ctrl C).

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 68 of 82

Go Back

Full Screen

Close

Quit

FTP clients

- command-based simple ftp client: `ftp`
- a command-based ftp client with more features: `ncftp`
- most browsers, using the FTP URL syntax
  e.g ftp://username@hostname/path/to/directory
- file/URL retrieval programs: e.g wget

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 69 of 82

Go Back

Full Screen

Close

Quit

FTP server

- ftp daemon (server) is the program which listens for ftp requests and executes ftp commands on remote machines
- it can be run standalone or from the INETD superserver
- ftp servers also support anonymous ftp, which is for public distribution of files without user authentication
- examples of ftp daemon programs: proftpd, wu-ftpd

proftpd server

- configuration files
  - `/etc/proftpd.conf` is the ftpd configuration file. One can specify maximum number of concurrent connections, authentication options and per-directory permissions among other things
  - `/etc/ftpusers` is the file which lists who are not allowed ftp access
- Anonymous ftp:
  - `/var/ftp/` or `/home/ftp` are the typical locations for the anonymous ftp directory
  - since the anonymous users are not authenticated, they are restricted to this directory tree only
  - they don't have access to full set of ftp commands

Secure File Transfer Program

- sftp (secure ftp) is an interactive file transfer program, which provides functionality similar to ftp
- it operates on top of the ssh transport, which means that the communication is encrypted
- it also provides other features of ssh, such as public key authentication and compression.
- sftp Example

```
$ stfp ajanshu@storage.it.iitb.ac.in
```

Secure Shell (SSH)

Secure Shell (SSH)

- ssh is a program for logging into a remote machine and executing commands on it
- provides secure encrypted communication between two trusted/untrusted hosts
- even X11 connections and arbitrary TCP/IP ports can be forwarded on the secure channel provided by ssh
- it provides functionality similar to rsh/rlogin, but in a secure way
- example- for remote login: `$ ssh hetanshu@it.iitb.ac.in`

SSH Authentication

- Host-based authentication: logging in is permitted if
  - the remote hostname is listed in /etc/hosts.equiv or /etc/ssh/shosts.equiv (on the remote machine where login is attempted)
  - and the usernames are the same on both sides
- RSA-based Host authentication:
  login is permitted similar to previous case, but only if the remote host's key can be varified by the server
- CAUTION: host-based authentication is inherently insecure, and should be avoided as a generel rule!

SSH Authentication (contd . . .)

- User Authentication:
  - Public-Key authentication
    - based on public-key cryptography:
    - there is a pair of complimentary keys for encryption and decryption respectively
    - it not possible (in realistic time, actually) to derive the decryption key from the encryption key
    - only the user has the private key, and the server has access to public key

File SHaring

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 77 of 82

Go Back

Full Screen

Close

Quit

File-Sharing

- Two important file-sharing protocols: SMB and NFS
  - SMB is the protocol for MS-Windows file sharing
  - “samba” is the package for using SMB on GNU/Linux
  - SMB also allows other services like printing etc.
  - `smbd` and `nmbd` are the daemon programs to share resources
  - smbclient and smbmount are used to use SMB shares
  - NFS is the protocol for Network filesystem

Misc Utilities

Useful Tools

- finger - user information lookup program
  #finger foo
  #finger foo@bar

- talk - talk to another user
  #talk foo
  #talk foo@domain.com

- telnet - user interface to communicate with another host using TELNET protocol.
  #telnet [domain name]

Resources

- manpages and documentation for all packages discussed
- Email:
  - Mail-Administrator-HOWTO
  - Sendmail Installation and Operation Guide
- DNS and BIND
  - DNS-HOWTO
  - Bind Operations Guide
- Firewalls
  - Firewall-HOWTO
  - Packet-Filtering HOWTO

Thanks..


GNU/LINUX
WORKSHOP