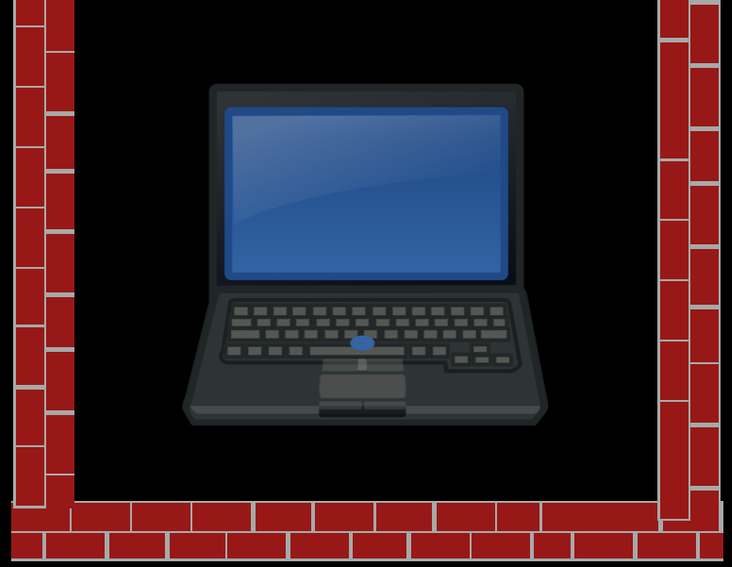
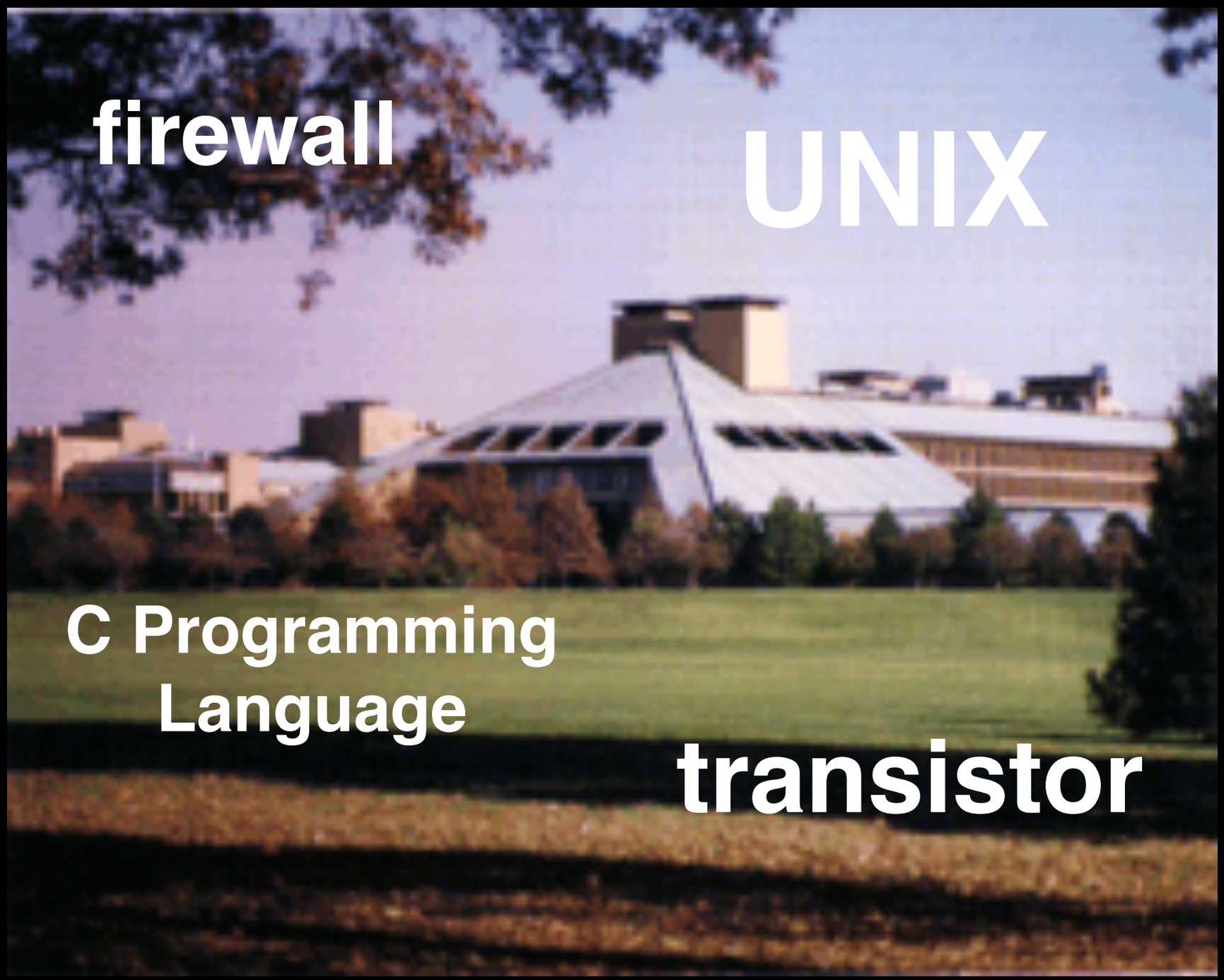
Firewalls

Network Security I



packet filtering

Bell Laboratories



packet filtering

network addresses
protocol
ports

TCP pseudo-header for checksum computation (IPv4)

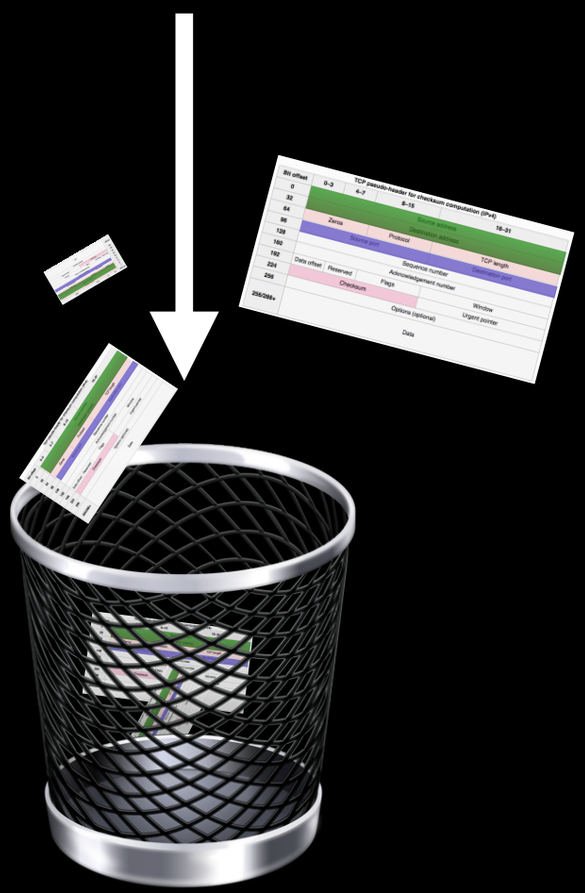
Bit offset	0–3	4–7	8–15	16–31
0	Source address			
32	Destination address			
64	Zeros		Protocol	TCP length
96	Source port			Destination port
128	Sequence number			
160	Acknowledgement number			
192	Data offset	Reserved	Flags	Window
224	Checksum			Urgent pointer
256	Options (optional)			
256/288+	Data			

silent

network addresses
protocol
ports

receiver

sender

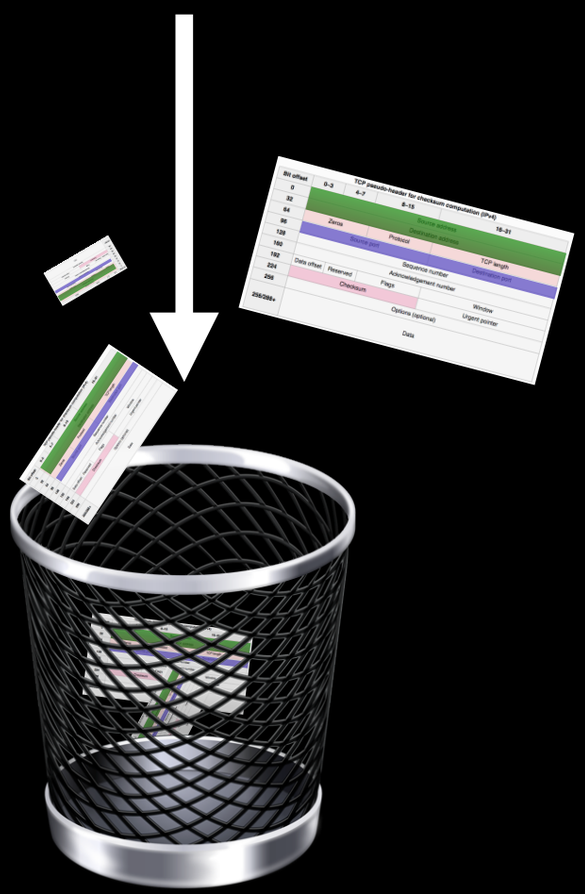


error response

network addresses
protocol
ports

receiver

sender



protocol

tcp
http
smtp
udp
pop
ftp
ssh
smtp

ports

22
80
110
546
21
156
666
1080

network address

192.168.0.21
192.168.0.39
192.168.0.101
192.168.1.45
192.168.1.4
192.168.0.221

Stateful Firewall

- maintains table of active connections
- context now helps filtering decisions
- table is not infinite, so connections will timeout and be cleared

- fast with table lookup
- overhead associated with establishing connection, but otherwise fast with table lookup implementation
- TCP - SYN, SYN-ACK, ACK => FIN, ACK
- UDP, SMPT - timeout only

host-based firewall

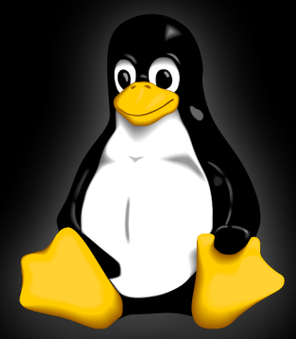
network firewall

network firewall

hardware implemented
software configured

host-based firewall

software implemented



iptables

Internet **C**onnection
Firewall

firewall

iptables

- administration tool for IPv4 filtering and NAT
- set up, maintain, and inspect tables of the IPv4 packet filter rules in the linux kernel
- table => chain = list of rules

targets

- firewall rule specifies criteria for packet and a target
- if packet does not match, next rule in chain is examined
- if match then the next rule specified by the value of the target, which can be name of chain or a value ACCEPT, DROP, QUEUE, or RETURN

ACCEPT

let packet through

QUEUE

pass packet to user space. different linux kernels handle with different queue handler implementations.

DROP

drop packet on the floor

RETURN

stop traversing this chain, resume at next rule in the previous (calling) chain. if end of built in chain is reached the target specified by the chain policy determines the fate

tables

tables

- the tables present at any time is dictated by kernel configuration options
- CHAINS

INPUT - packets destined to local sockets

FORWARD - packets being routed through box

OUTPUT - packets generated by local process

PREROUTING - alter incoming packets before routing

POSTROUTING - alteration of packets before they go out

raw

- configure exemptions from connection tracking with NOTRACK target
- registers as higher priority, called before ip_conntrack, or other ip tables
- PREROUTING, OUTPUT

filter

- default table
- INPUT, FORWARD, OUTPUT

nat

- consulted when a packet that creates a new connection is encountered

- PREROUTING, OUTPUT, POSTROUTING

mangle

- specialized packet alteration

- PREROUTING, OUTPUT, INPUT, FORWARD, POSTROUTING

security

- used for mandatory access control network rules
- SECMARK, CONNSECMARK marks

  SECMARK matches an entry in table applying a label that can be used to enforce a policy on packet

  CONSECMARK marks all packets within session with a label that can be used to enforce a policy

- INPUT, OUTPUT, FORWARD

rules

rulesets

- order is very important
- rules exist in memory and can be lost if not saved (iptables-save)
- making a script is very helpful so if things get flushed (iptables —flush) there is an easy recovery
- hint: if on ssh, don’t lock yourself out!

first rule example

- iptables -A INPUT -s 192.168.1.100 -d 192.168.1.10 -p tcp —dport 22 -j ACCEPT

  -A => ‘append’ this rule to input chain

  -s => source address

  -d => destination address (server connecting to)

  -p => protocol

  —dport => destination port

  -j => jump, if everything in this rule matches then accept

Bash Script:

Clear all filter table rules

Allow SSH

Allow all traffic for local subnetwork 192.XXX.XXX.XXX

Allow Ping Requests

Allow DNS Traffic (TCP and UDP)

Allow HTTP/HTTPS traffic

Allow FTP Traffic

Deny all other traffic

Hint: What does OUTPUT and INPUT mean for iptables? Also try to attempt only using man pages for reference.

conclusion

- firewalls do not protect against viruses
- simple gateway
- linux configuration using the iptables tool