Announcements

- Project #6
  - Clarifications & corrections available on web
- Reading Chapter 15 (Networks)

Viruses

- Most common on systems with little security
  - easy to write to boot blocks, system software
  - never run untrusted software with special privileges
  - Don’t perform daily operations with root/system privileges
- Possible to write system independent viruses
  - MS Word virus
    - uses macros to call into the OS
  - HTML (javascript)
  - Flash

Project #6 Notes

- Uid
  - First process has uid of 0
  - Spawned processes
    - Inherit uid of parent
    - Unless setuid bit is set on program to run, then the uid of the owner of that file is used
- ACLs
  - First ACL entry is owner
  - Others are for other users
    - Can delete these entires with setACl(file, uid, 0)
  - Uid 0 can open any file regardless of ACLs

Access Matrix

- Abstraction of protection for objects in a system.
  - Rows are domains (users or groups of users)
  - Columns are objects (files, printers, etc.)
  - Items are methods permitted by a domain on an objects
    - read, write, execute, print, delete, …
- Representing the Table
  - simple representation (dense matrix) is large
  - sparse representation possible: each non-zero in the matrix
  - observation: same column used frequently
    - represent groups of users with a name and just store that
  - create a default policy for some objects without a value
- Revocation of access
  - when are access rights checked?
  - selective revocation vs. global

Access Matrix

	F1	F2	F3	Laser Printer	
D1	read		execute		
D2			execute	print	
D3	read, write		execute		
D4			execute		
D5		delete			

- Rows represent users or groups of users
- Columns represent files, printers, etc.

Capabilities

- Un-forgeable Key to access something
- Implementation: a string
  - I.e. a long numeric sequence for a copier
- Implementation: A protected memory region
    - tag memory (or procedures) with access rights
      - example - x86 call gate abstraction
    - permit rights amplification

Monitoring

- Record (log) significant events
  - attempts to login to the system
  - changes to selected files or directories
- Possible to compromise the log
  - the user or software breaking in could delete all or part of the logs
  - could record logs to non-erasable storage
    - have a line printer attached to the machine
    - use DVD-ROM drives
  - send data to a secure remote host

Tripwire

- Compute a set of expectorations about system
  - Hash of file contents
  - Dates on files
- Store database of values
  - On read-only media
  - Offline
- Periodically
  - Compare database to current system
  - Report any differences

Encryption: protecting info from being read

- Given a message m
  - use a key k, and function E_k to compute $E_k(m)$
  - store or send only $E_k(m)$
  - use a second second key k and function $D_{k'}$ such that
    - $D_{k'}(E_k(m)) = m$
  - E_k and $D_{k'}$ need not be kept a secrete
- If k=k' it's called private key encryption
  - need to keep k secret
  - example AES-256
- if k != k', it's called public key encryption
  - need only keep one of them secret
  - if k' is secret, anyone can send a private message
  - if k is secret, it is possible to "sign" a message
  - still need a way to authenticate k or k' for a user
  - example RSA

Public Key Encryption

- Split into public and private keys
  - public key used to encrypt messages
    - publish this key widely
  - private key used to decrypt messages
    - keep this key a secret
- RSA
  - algorithm for computing public/private key pairs
  - based on problems involved in factoring large primes
  - for an n bit message P, C = (P^e mod n), and P = (C^d mod n)
- Other Public Key Algorithms
  - knapsack
    - given a large collection of objects with different weights
    - public key is the total weight of a subset of the objects
    - private key is the list of objects

One Time Pad

- Key Idea: randomness in key
- Create a random string as long as the message
  - each party has the pad
  - xor each bit of the message with the a bit of the key
- Almost impossible to break
- Some practical problems
  - need to ensure key is not captured
  - a one bit drop will corrupt the rest of the message

Secure Socket Layer

- Goal:
  - Provide secure access to remote services
  - Authenticate remote servers to local users
  - Allow remote systems to authenticate users
  - Permit encrypted communication
- Approach
  - Public Key Cryptography
    - Certificates (signed by certificate authorities)
  - Sever sends:
    - Certificate (signed use CA's private key)
    - Certificate contains server's public key
    - Client responds by encrypting reply using servers pub key
    - Server checks response with private key

Sending Data

- Data is split into *packets*
  - limited size units of sending information
  - can be
    - fixed sized (ATM)
    - variable size (Ethernet)
- Need to provide a destination for the packet
  - need to identify two levels of information
    - machine to send data to
    - comm abstraction (e.g. process) to get data
  - address may be:
    - a globally unique destination
      - for example every host has a unique id
    - may unique between hops
      - unique id between two switches

TCP/IP Protocol

- Name for a family of Network and Transport layers
  - can run over many link layers:
    - Arpanet, Ethernet, Token Ring, SLIP/PPP, T1/T3, etc.
- IP - Internet Protocol
  - network level packet oriented protocol
  - 32 bit host addresses (dotted quad 128.8.128.84)
  - 8 bit protocol field (e.g. TCP, UDP, ICMP)
- TCP - Transmission Control Protocol
  - transport protocol
  - end-to-end reliable byte streams
  - provides ports for application specific end-points
- UDP- user datagram protocol
  - transport protocol
  - unreliable packet service
  - provides ports for application specific end-points

TCP/IP History

- Arpanet was the origin of today's Internet
  - started in 1969 to connect universities and DoD sites
  - early example of packet switched network
  - original links were 64kbps and 9.6kpbs
- TCP/IP v4
  - started in use Jan 1, 1983
  - This was a *flag day*
    - all systems had to change to the new protocol at once
    - with the modern Internet this would be **hard** to do
- TCP/IP v6
  - Moves to 128 bit addresses
  - Simplified packet header

Subnet Addressing

- Single site which has many physical networks
  - Only local routers know about all the physical nets
  - Site chooses part of address that distinguishes between physical networks
- subnet mask: splits the IP address into two parts
  - /xx notation defines boundary where xx is the number of bits in part 1
  - First part is network mask
  - Second part is address within that network
- Common /24 site mask 255.255.255.0
  - use 24 bits represent physical net
  - Final 8 bits represent host

Routing

- How does a packet find its destination?
  - problem is called routing
- Several options:
  - source routing
    - end points know how to get everywhere
    - each packet is given a list of hops before it is sent
  - hop-by-hop
    - each host knows for each destination how to get one more hop in the right direction
- Can route packets:
  - per session
    - each packet in a connection takes same path
  - per packet
    - packets may take different routes
    - possible to have out of order delivery

Routing IP Datagrams

- Direct Delivery:
  - a machine on a physical network can send a physical frame directly to another
  - transmission of an IP datagram between two machines on a single physical network does not involve routers.
    - Sender encapsulates datagram into a physical frame, maps destination IP address to a physical address and sends frame directly to destination
  - Sender knows that a machine is on a directly connected network
    - compare network portion of destination ID with own ID - if these match, the datagram can be sent directly
  - Direct delivery can be viewed as the final step in any datagram transmission

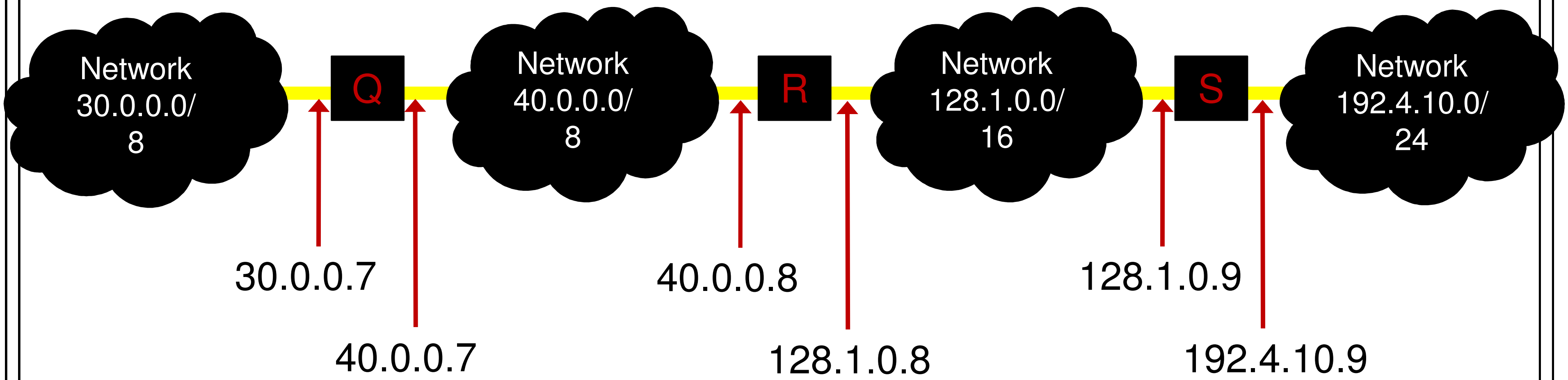
Routing Datagrams (cont.)

- Indirect Delivery
  - sender must identify a router to which a datagram can be sent
  - sending processor can reach a router on the sending processor's physical network (otherwise the network is isolated!)
  - when frame reaches router, router extracts encapsulated datagram and IP software selects the next router
    - datagram is placed in a frame and sent off to the next router

Table Driven Routing

- Routing tables on each machine store information about possible destinations and how to reach them
- Routing tables only need to contain network prefixes, not full IP addresses
  - No need to include information about specific hosts
- Each entry in a routing table points to a router that can be reached across a single network
- Hosts and routers decide
  - can packet be directly sent?
  - which router should be responsible for a packet (if there is more than one on physical net)

Routing (w/ subnets)



To reach hosts on network	Mask*	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	<DIRECT>
128.1.0.0	255.255.0.0	<DIRECT>
192.4.10.0	255.255.255.0	128.1.0.9

Mask field is used to extract the network part of an address during lookup.

If((Mask[i] & D) == Destination[i]) forward to nextHop[i]

Consider a datagram destined for address 192.4.10.3 and the datagram arrives at router R

Extract destination IP address, D from datagram and compute network prefix N

255.0.0.0&192.4.10.3 is not equal to 30.0.0.0

<same for entry 2 and 3>

255.255.255.0&192.4.10.3=192.4.10.0
→ send to 128.1.0.9

Example from Comer book: Internetworking with TCP/IP: volume 1 [Third Edition]

21

Algorithm: RouteDatagram (Datagram, RoutingTable)

Extract destination IP address, D, from datagram and compute network prefix N

If N matches any directly connected network address
  [Direct delivery]

Else if the table contains a host-specific route for D
  [send datagram to next-hop specified in table]

Else if the table contains a route for network N
  [send datagram to next-hop specified in table]

Else if the table contains a default route
  [send the datagram to the default route]

Else *declare a routing error*

Algorithm from Comer book: Internetworking with TCP/IP: volume 1 [Third Edition]

22