Denial of Service

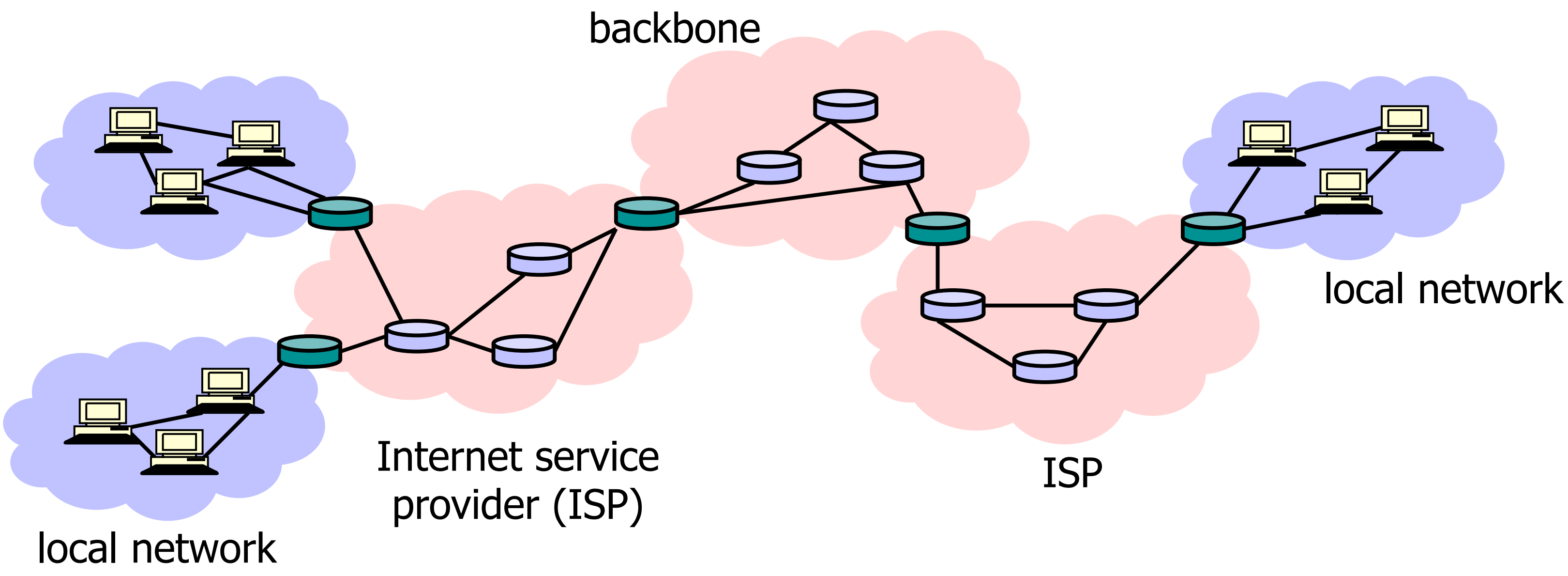
EJ Jung

Denial of Service (DoS)

- Goal: overwhelm victim machine and deny service to its legitimate clients

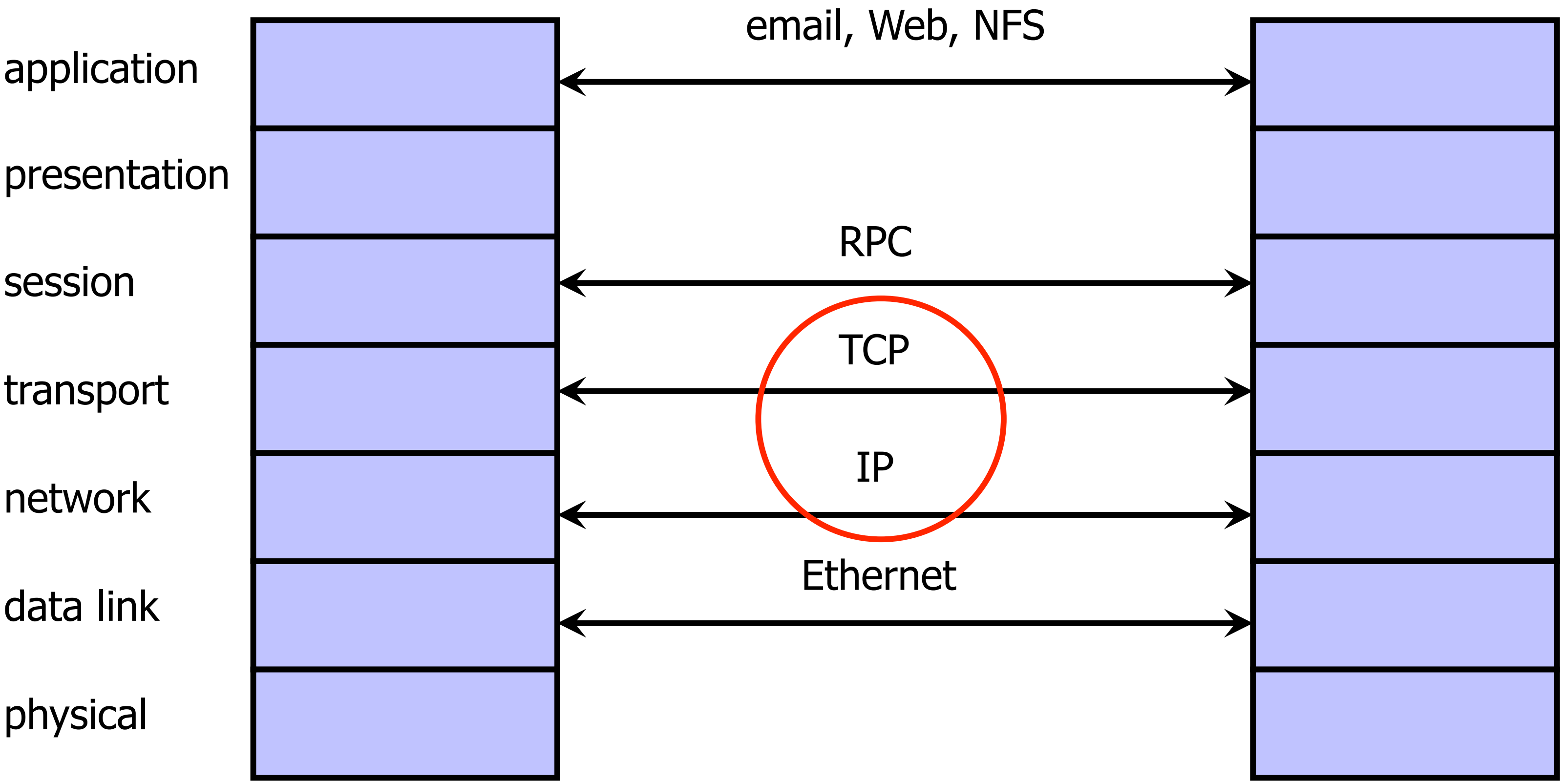
- Exploits vulnerabilities in many network protocols, anywhere in the stack

Internet Infrastructure

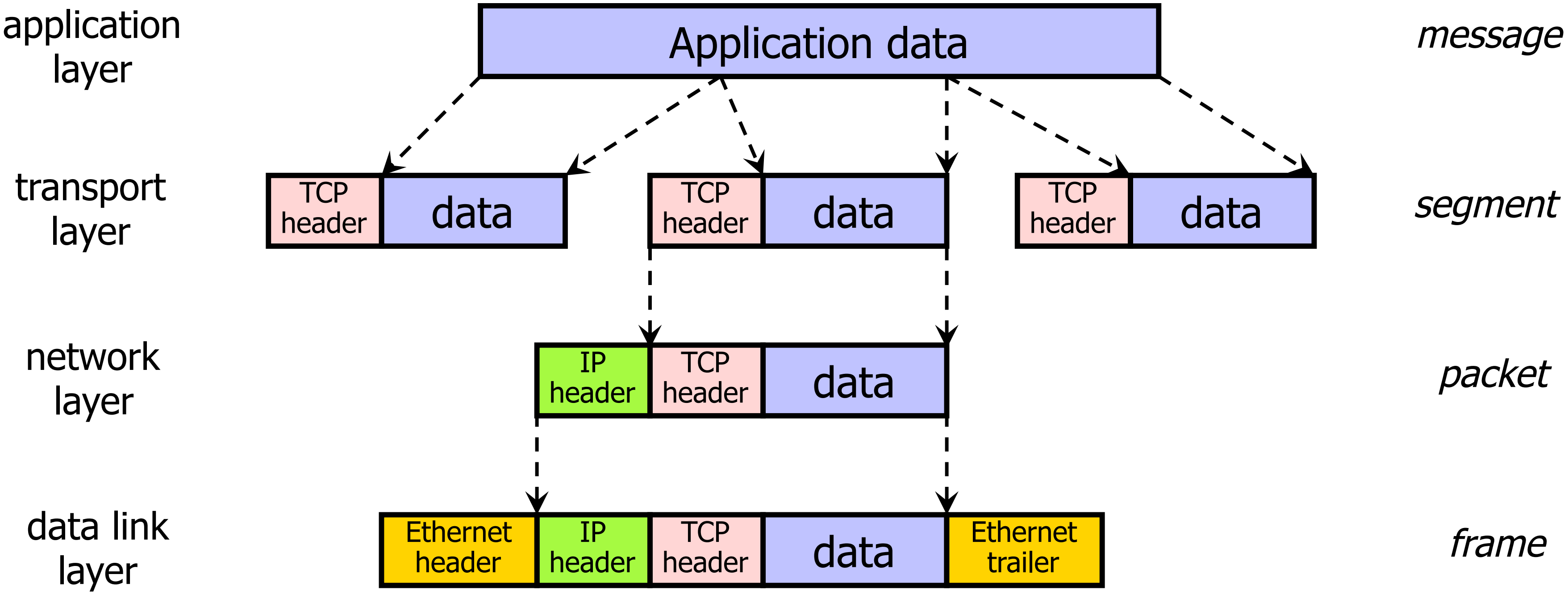


- TCP/IP for packet routing and connections
- Border Gateway Protocol (BGP) for route discovery
- Domain Name System (DNS) for IP address discovery

OSI Protocol Stack



Data Formats



application layer

Application data

message

transport layer

TCP header | data

TCP header | data

TCP header | data

segment

network layer

IP header | TCP header | data

packet

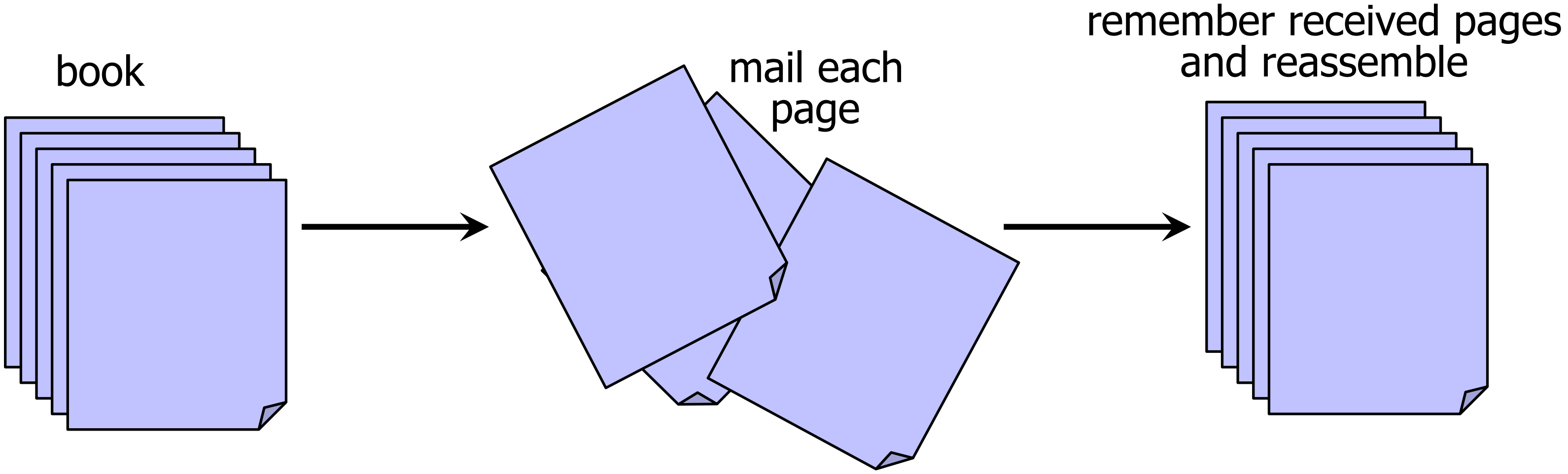
data link layer

Ethernet header | IP header | TCP header | data | Ethernet trailer

frame

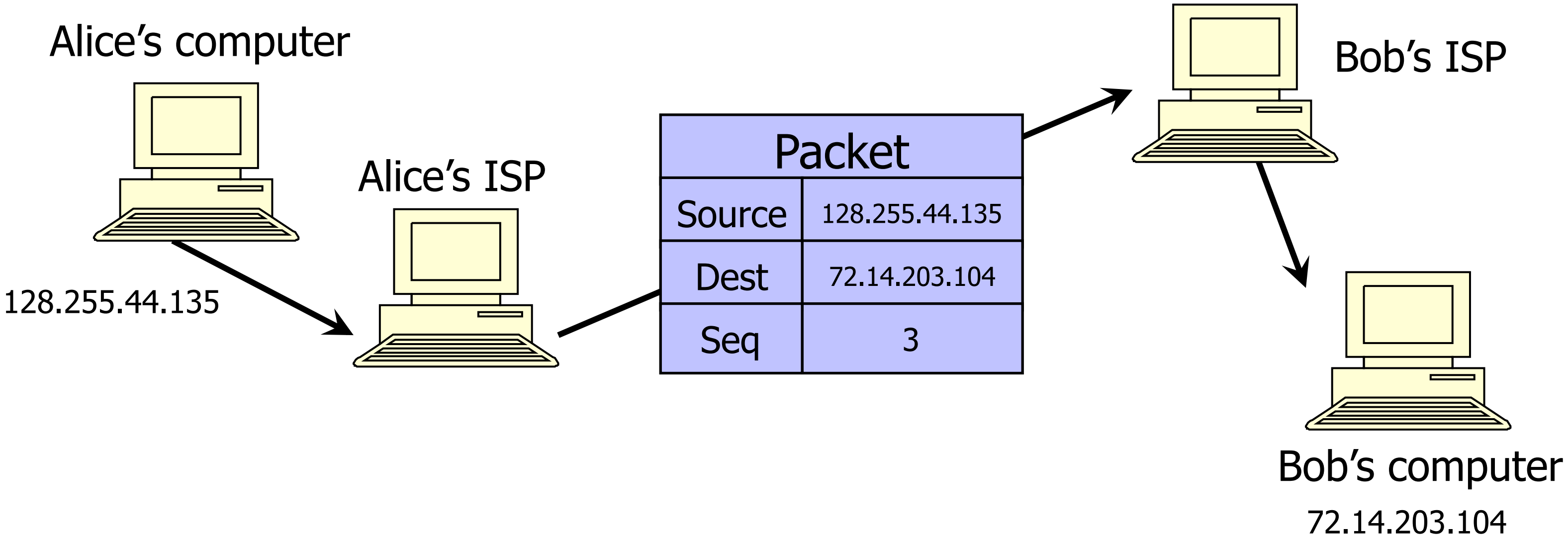
TCP (Transmission Control Protocol)

- Sender: break data into packets
  - Sequence number is attached to every packet
- Receiver: reassemble packets in correct order
  - Acknowledge receipt; lost packets are re-sent
- Connection state maintained on both sides



IP (Internet Protocol)

- Connectionless
  - Unreliable, “best-effort” protocol
- Uses numeric addresses for routing
  - Typically several hops in the route



Packet	
Source	128.255.44.135
Dest	72.14.203.104
Seq	3

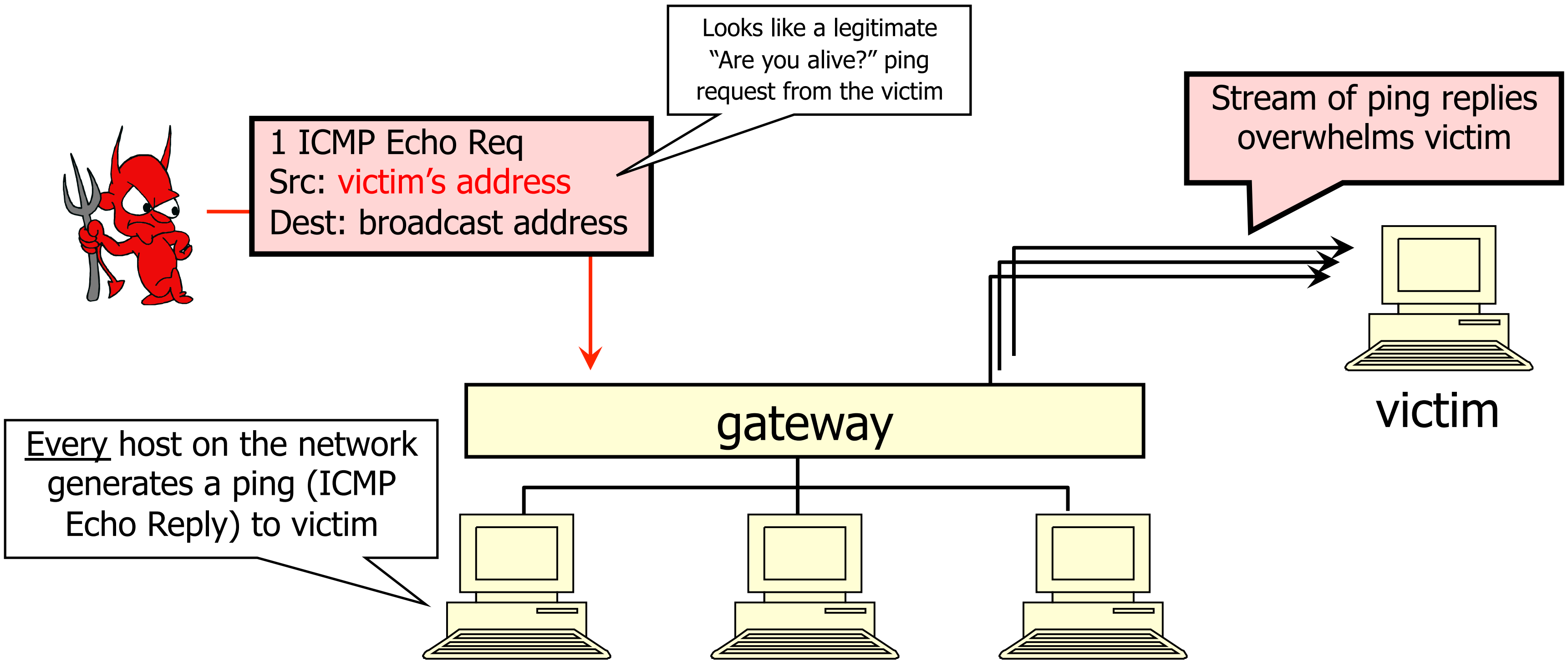
ICMP (Internet Control Message Protocol)

- Extension of IP by RFC 792
- Provides feedback about network operation
  - "Out-of-band" messages carried in IP packets
  - Error reporting, congestion control, reachability, etc.
- Example messages:
  - Destination unreachable
  - Time exceeded
  - Redirect to better gateway
  - Reachability test (echo / echo reply)
  - Message transit delay (timestamp request / reply)

DoS attacks in TCP/IP

- IP addresses are public
  - Smurf attacks
- TCP connection requires state
  - SYN flooding
- TCP state is easy to guess
  - TCP spoofing and connection hijacking

Smurf Attack



Looks like a legitimate "Are you alive?" ping request from the victim

Stream of ping replies overwhelms victim

1 ICMP Echo Req
Src: victim's address
Dest: broadcast address

Every host on the network generates a ping (ICMP Echo Reply) to victim

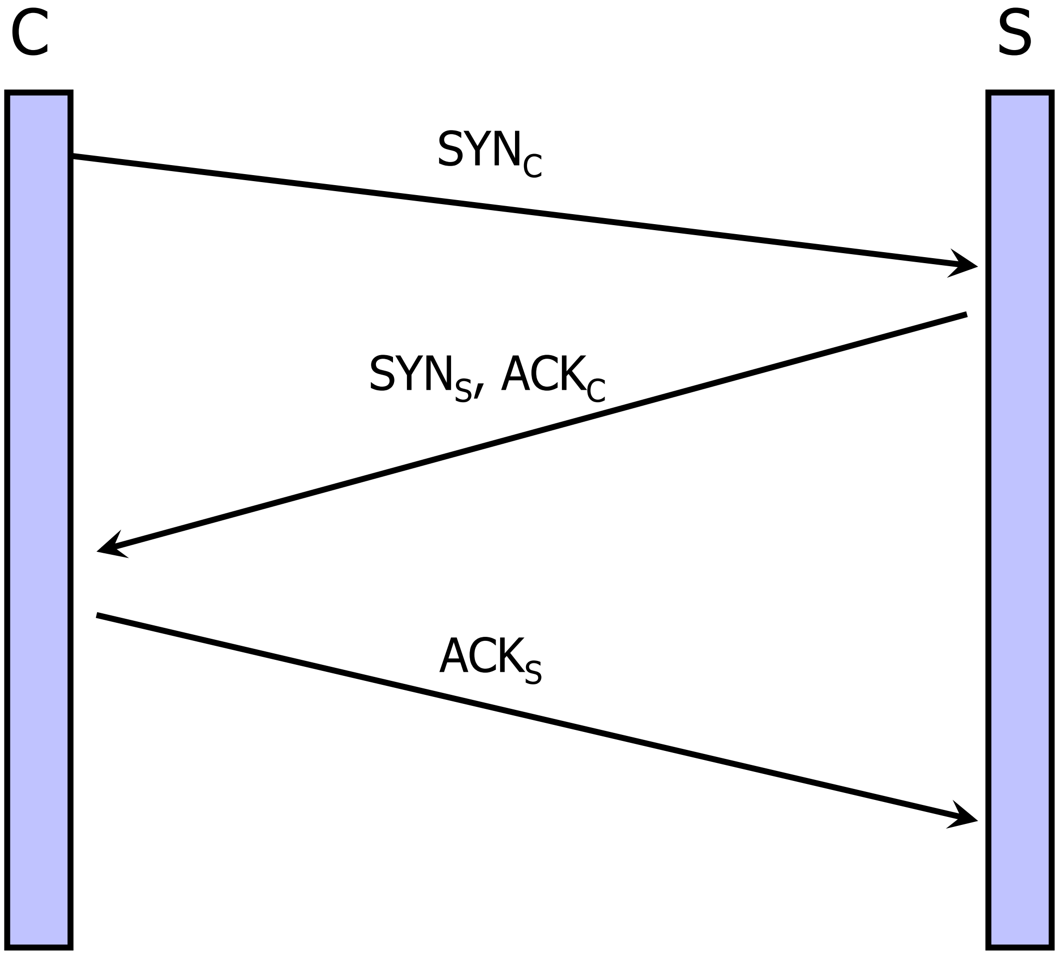
Solution: reject external packets to broadcast addresses at router

“Ping of Death”

- If an old Windows machine received an ICMP packet with a payload longer than 64K, machine would crash or reboot
  - Programming error in older versions of Windows
  - Packets of this length are illegal, so programmers of Windows code did not account for them

Solution: patch OS, filter out ICMP packets

TCP Handshake



C

S

SYN_C

SYN_S, ACK_C

ACK_S

Listening…

Store data

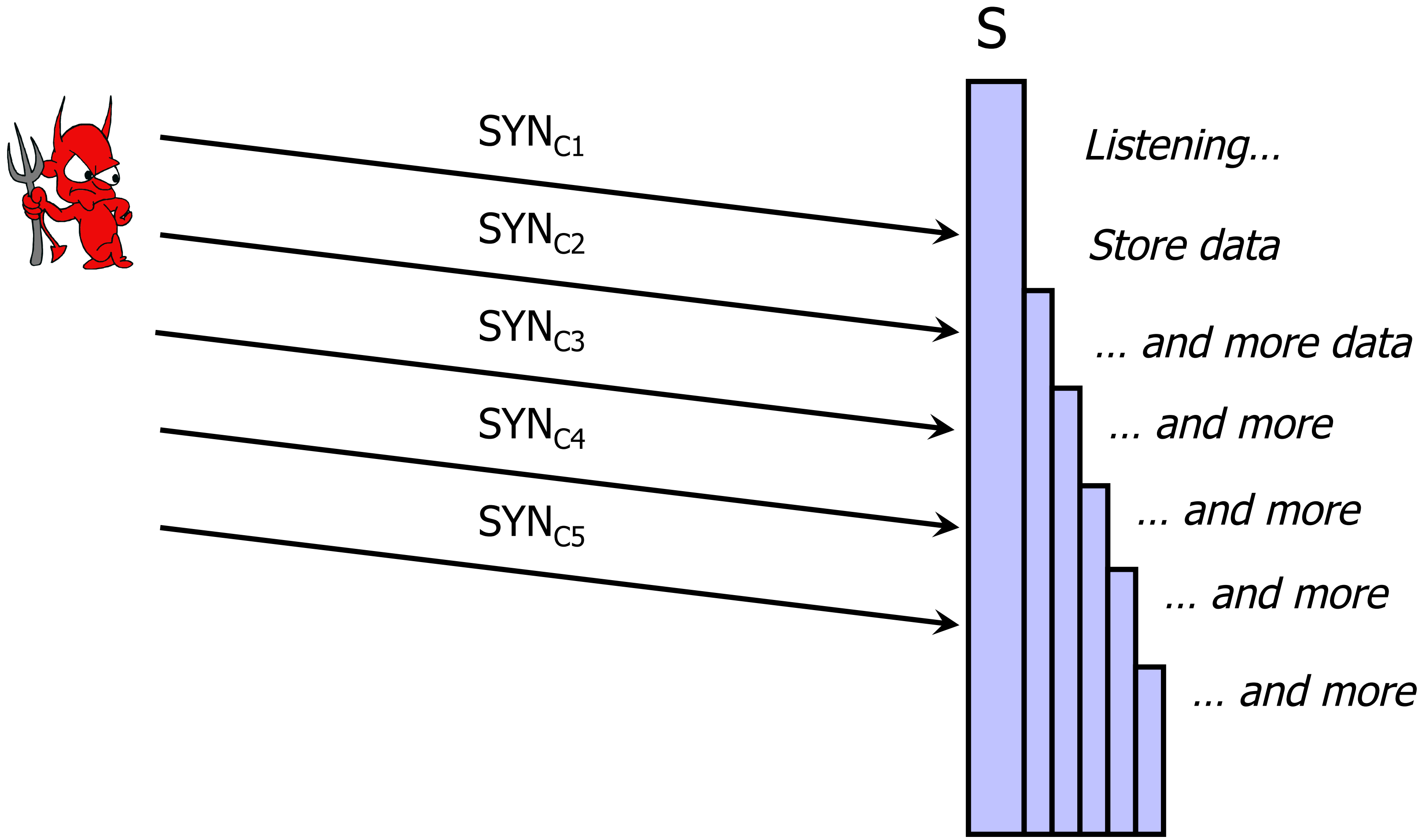
(connection state, etc.)

Wait

Connected

UNIVERSITY of SAN FRANCISCO
department of computer science

SYN Flooding Attack



SYN Flooding Explained

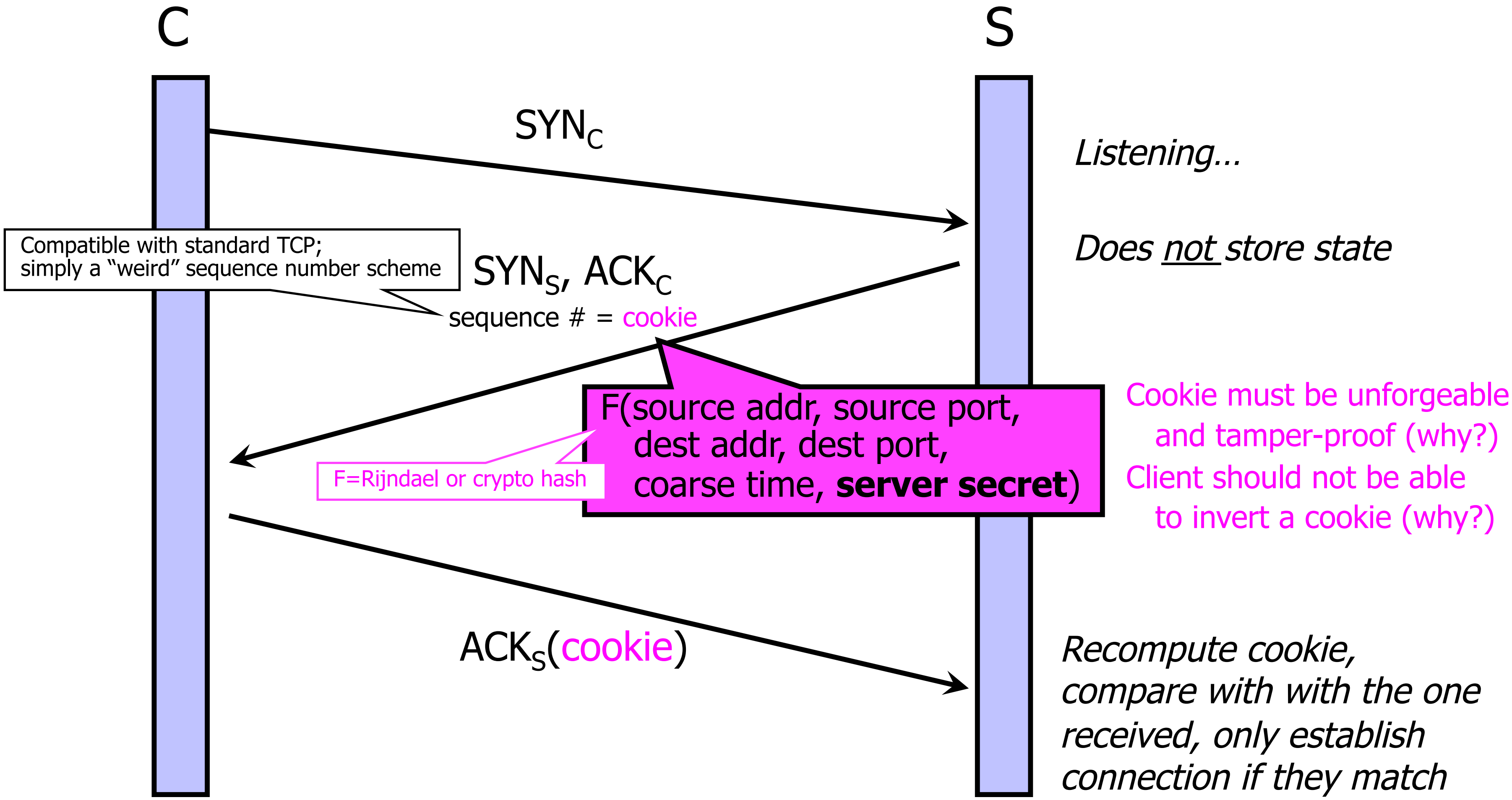
- Attacker sends many connection requests with spoofed source addresses
- Victim allocates resources for each request
  - Connection state maintained until timeout
  - Fixed bound on half-open connections
- Once resources exhausted, requests from legitimate clients are denied
- This is a classic denial of service (DoS) attack
  - Common pattern: it costs nothing to TCP initiator to send a connection request, but TCP responder must allocate state for each request (asymmetry!)

Preventing Denial of Service

- DoS is caused by asymmetric state allocation
  - If responder opens a state for each connection attempt, attacker can initiate thousands of connections from bogus or forged IP addresses
- Cookies ensure that the responder is stateless until initiator produced at least 2 messages
  - Responder’s state (IP addresses and ports of the connection) is stored in a cookie and sent to initiator
  - After initiator responds, cookie is regenerated and compared with the cookie returned by the initiator

SYN Cookies

[Bernstein & Schenk]



C

S

SYN_C

Listening…

Does <u>not</u> store state

Compatible with standard TCP; simply a “weird” sequence number scheme

SYN_S, ACK_C

sequence # = cookie

F(source addr, source port, dest addr, dest port, coarse time, **server secret**)

F=Rijndael or crypto hash

Cookie must be unforgeable and tamper-proof (why?)

Client should not be able to invert a cookie (why?)

ACK_S(cookie)

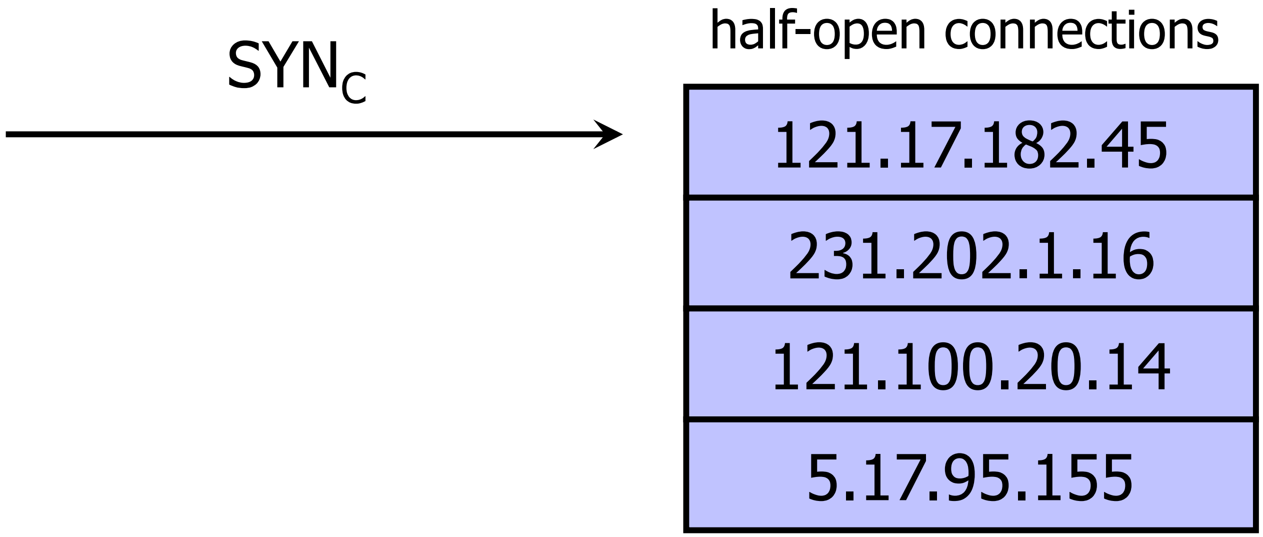
Recompute cookie, compare with with the one received, only establish connection if they match

More info: http://cr.yp.to/syncookies.html

Anti-Spoofing Cookies: Basic Pattern

- Client sends request (message #1) to server
- Typical protocol:
  - Server sets up connection, responds with message #2
  - Client may complete session or not (potential DoS)
- Cookie version:
  - Server sends hashed connection data back
    - Send message #2 later, after client confirms he is listening
  - Client confirms by returning hashed data
    - If source IP address is bogus, attacker can't confirm
  - Need an extra step to send postponed message #2
    - Ok in TCP since the extra step (SYN-ACK) is already there

Another Defense: Random Deletion



SYN_C

half-open connections

121.17.182.45

231.202.1.16

121.100.20.14

5.17.95.155

- If SYN queue is full, delete random entry
    - Legitimate connections have a chance to complete
    - Fake addresses will be eventually deleted
- Easy to implement

TCP Connection Spoofing

- Each TCP connection has an associated state
  - Sequence number, port number
- TCP state is easy to guess
  - Port numbers are standard, sequence numbers are often predictable
  - Can inject packets into existing connections
- If attacker knows initial sequence number and amount of traffic, can guess likely current number
  - Send a flood of packets with likely sequence numbers

DoS by Connection Reset

- If attacker can guess current sequence number for an existing connection, can send Reset packet to close it
  - With 32-bit sequence numbers, probability of guessing correctly is $1/2^{32}$ (not practical)
  - Most systems accept large windows of sequence numbers $\Rightarrow$ much higher probability of success
    - Need large windows to handle massive packet losses
- Especially effective against long-lived connections
  - For example, BGP (Border Gateway Protocol)

User Datagram Protocol (UDP)

- UDP is a connectionless protocol
  - Simply send datagram to application process at the specified port of the IP address
  - Source port number provides return address
  - Applications: media streaming, broadcast
- No acknowledgement, no flow control, no message continuation
- Denial of service by UDP data flood

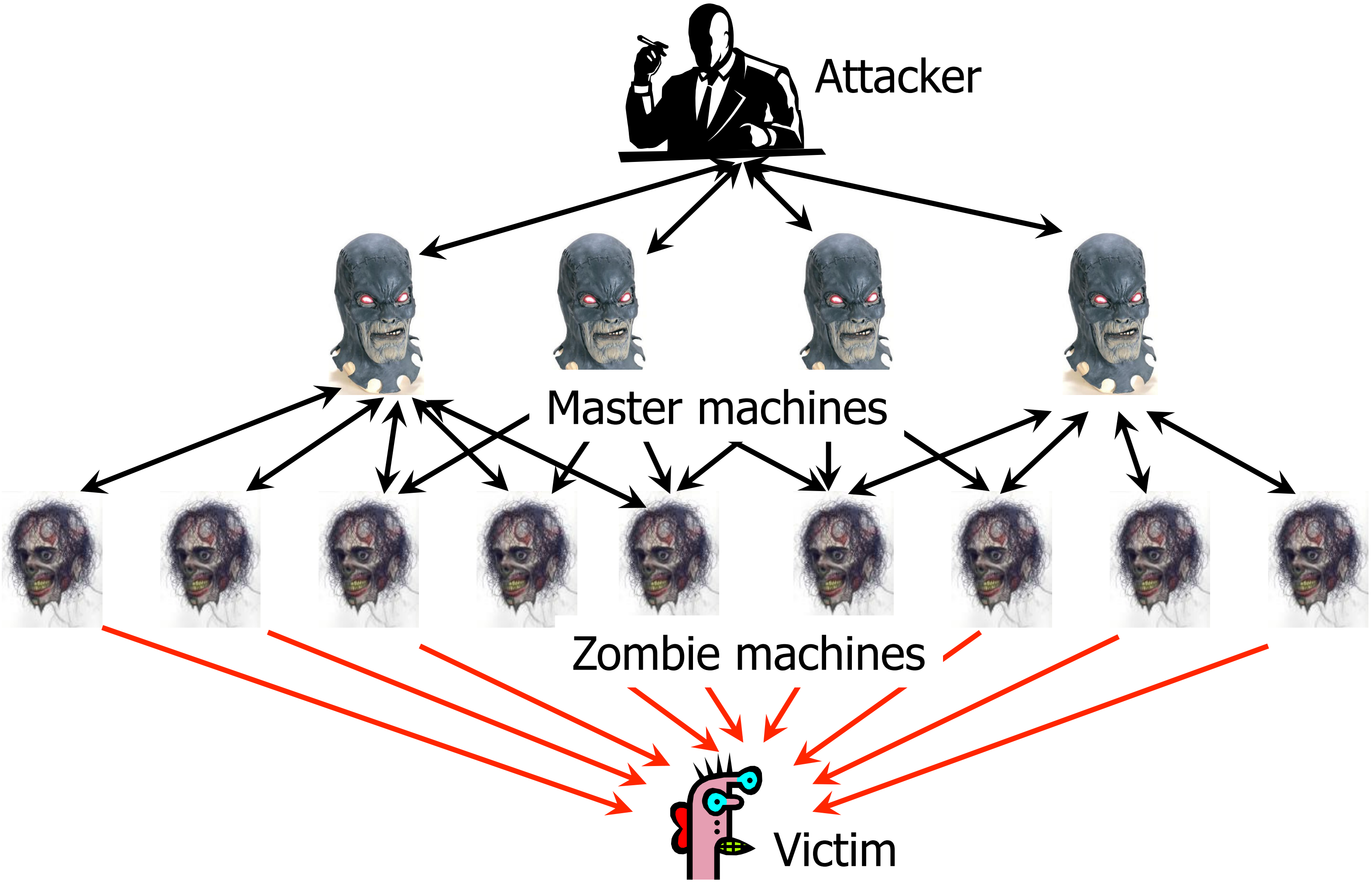
Countermeasures

- Above transport layer: SSL/TLS and SSH
  - Protects against connection hijacking and injected data
  - Does <u>not</u> protect against DoS by spoofed packets
- Above transport layer: Kerberos
  - Provides authentication, protects against spoofing
  - Does <u>not</u> protect against connection hijacking
- Network (IP) layer: IPSec
  - Protects against hijacking, injection, DoS using connection resets, IP address spoofing

Distributed Denial of Service (DDoS)

- First, scan hundreds of thousands of computers on the Internet for known vulnerabilities
- Turn vulnerable computers into “zombies”
  - Exploit vulnerabilities to gain root access, install attack and communication tools, use them for further scans
- Form a distributed attack network from zombies
  - Choose a subset of compromised machines with desired network topology and characteristics
- Command zombies to stage a coordinated attack on the victim

DDoS Architecture



DDoS Tools: Trin00

- Scan for known buffer overflows in Linux & Solaris
  - Unpatched versions of wu-ftpd, statd, amd, …
  - Root shell on compromised host returns confirmation
- Install attack daemon using remote shell access
- Send commands (victim IP, attack parameters, etc.), using plaintext passwords for authentication
  - Attacker to master: TCP, master to zombie: UDP
  - To avoid detection, daemon issues warning if someone connects when master is already authenticated
- In August of 1999, a network of 227 Trin00 zombies took U. of Minnesota offline for 3 days

DDoS Tools: Tribal Flood Network

- Supports multiple DoS attack types
  - Smurf; ICMP, SYN, UDP floods
- Attacker runs masters directly via root backdoor; masters talk to zombies using ICMP echo reply
  - No authentication of master's commands, but commands are encoded as 16-bit binary numbers inside ICMP packets to prevent accidental triggering
  - Vulnerable to connection hijacking and RST sniping
- List of zombie daemons' IP addresses is encrypted in later versions of TFN master scripts
  - Protects identities of zombies if master is discovered

DDoS Tools: Stacheldraht

- Combines "best" features of Trin00 and TFN
  - Multiple attack types (like TFN)
- Symmetric encryption for attacker-master connections
- Master daemons can be upgraded on demand
- February 2000: crippled Yahoo, eBay, Amazon, Schwab, E*Trade, CNN, Buy.com, ZDNet
  - Smurf-like attack on Yahoo consumed more than a Gigabit/sec of bandwidth
  - Sources of attack still unknown

U. of Toronto, 2004(from David Lie's slides)

- Date: Fri, 19 Mar 2004
- Quote from email:

  "The campus switches have been bombarded with these packets […] and apparently 3Com switches reset when they get these packets. This has caused the campus backbone to be up and down most of yesterday. The attack seems to start with connection attempts to port 1025 (Active Directory logon, which fails), then 6129 (DameWare backdoor, which fails), then 80 (which works as the 3Com's support a web server, which can't be disabled as far as we know). The HTTP command starts with 'SEARCH /\x90\x02\xb1\x02' […] then goes off into a continual pattern of '\x90' "

Defending Against DDoS

- Authenticate packet sources
  - Not feasible with current IP (unless IPSec is used)
- Filter incoming traffic on access routers or rate-limit certain traffic types (ICMP and SYN packets)
  - Need to correctly measure normal rates first!
- Force clients to do an expensive computation or to prove that they are human
  - If connection requested, ask client to solve a "puzzle"
    - E.g., invert a short hash value or solve a graphical Turing test
  - Honest clients can easily do this, but zombies can't
  - Requires modification of TCP/IP stack (not feasible?)

Finding Attack Sources

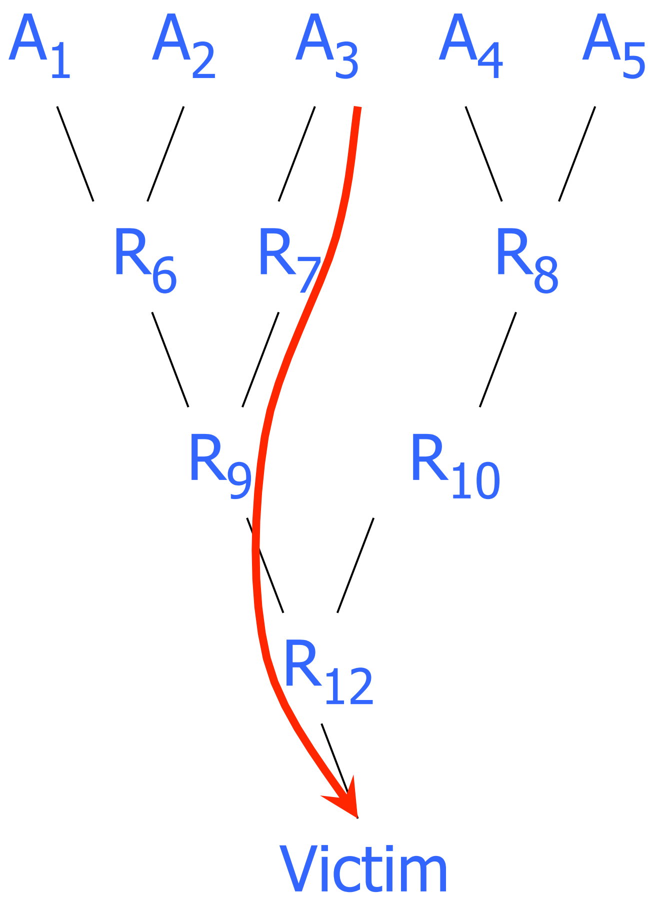
- Note: this will only locate zombies
  - Forensics on zombie machines can help find masters and the attacker who remotely controls them
- Can use existing IP routing infrastructure
  - Link testing (while attack is in progress)
  - Packet logging (for post-mortem path reconstruction)
- …or propose changes to routing infrastructure
  - IP traceback (e.g., via packet marking)
  - … and dozens of other proposals
  - Changing routing infrastructure is hard!

Link Testing

- Only works while attack is in progress
- Input debugging
  - Victim reports attack to upstream router
  - Router installs a filter for attack traffic, determines which upstream router originated it
  - Repeat upstream (requires inter-ISP cooperation)
- Controlled flooding
  - Iteratively flood each incoming link of the router; if attack traffic decreases, this must be the guilty link
    - Use a form of DoS to throttle DoS traffic (!!)
  - Need a good network map and router cooperation

IP Traceback Problem

- How to determine the path traversed by attack packets?
- Assumptions:
  - Most routers remain uncompromised
  - Attacker sends many packets
  - Route from attacker to victim remains relatively stable

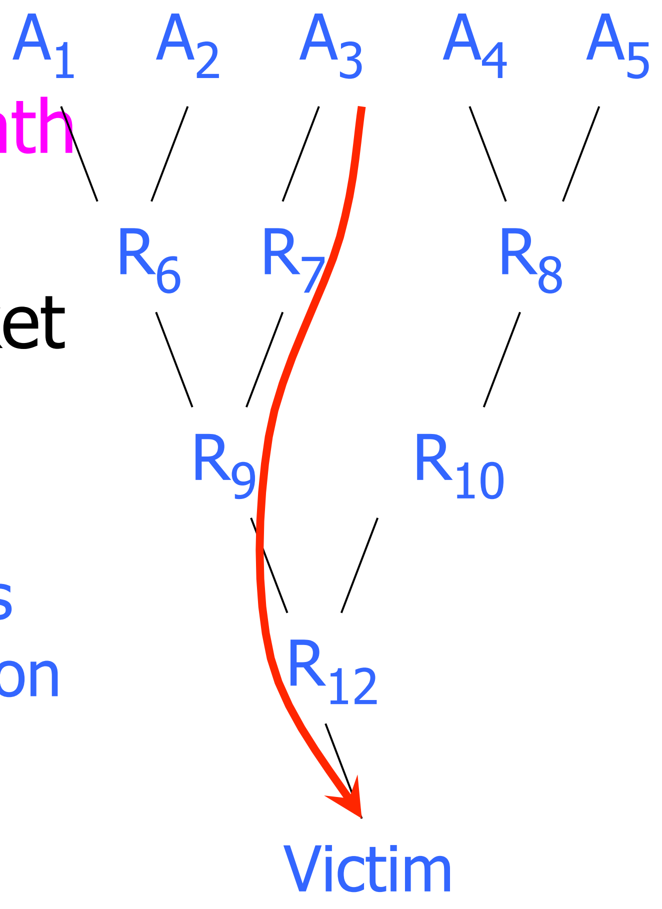


Obvious Solution Doesn’t Work

- Obvious solution: have each router on the path add its IP address to packet; victim will read path from the packet
- Problem: requires space in the packet
  - Paths can be long
  - Current IP format provides no extra fields to store path information
  - Changes to packet format are not feasible

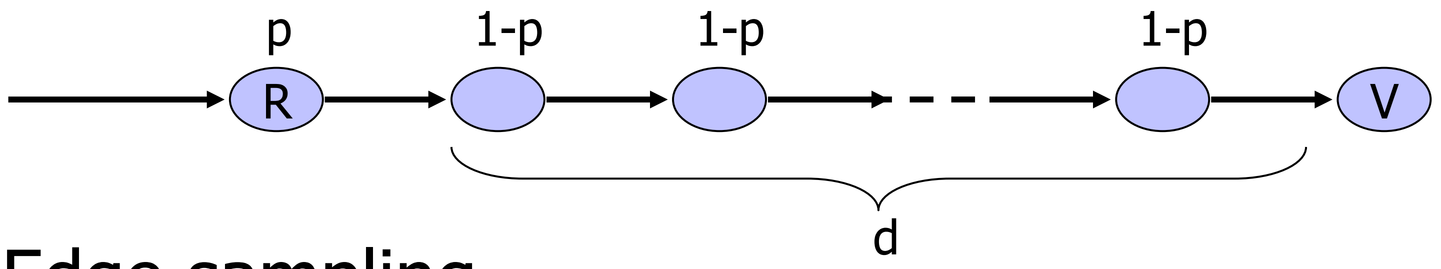
Probabilistic Packet Marking

- DDoS involves many packets on the same path
- With some probability, each router marks packet with router's address
  - Fixed space per packet
  - Large number of packets means that each router on the path will appear in some packet



Node and Edge Sampling

- Node sampling
  - With probability p, router stores its address in packet
  - Router at distance d shows up with probability $p(1-p)^d$



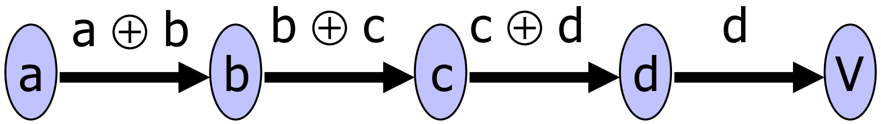
- Edge sampling
  - Packet stores an edge and distance since it was stored
    - More space per packet, but fewer packets to reconstruct path
  - With probability p, router stores the current edge and sets distance to 0, else increments distance by 1

Storing Edges in IP Packets

- 16-bit Identification field
  - Used for fragmentation
  - Fragmentation is rare
- Storing an edge in 16 bits

offset	distance	edge chunk
0 2	3 7	8 15

  - Store start⊕end
  - Work backwards to get path: (start⊕end)⊕end = start



Version	Header length
Type of service	
Total length	
Identification	
Flags	Fragment offset
Time to live	
Protocol	
Header checksum	
Source address of originating host	
Destination address of target host	
Options	
Padding	
IP Data	