Network Security

(From Slides/Notes)

Dina Katabi
dk@mit.edu
nms.csail.mit.edu/~dina

Network Attacks Are Common

- Attack Types:
  - Denial of service attacks
  - Spam
  - Worms & Viruses
  - and others
- Attack targets
  - End systems including attacks on Web servers, TCP, etc.
  - Links
  - Routers
  - DNS
  - And others
- Who are the attackers?
  - Script kiddies
  - Professionals who do it for money

Mounting An Attack

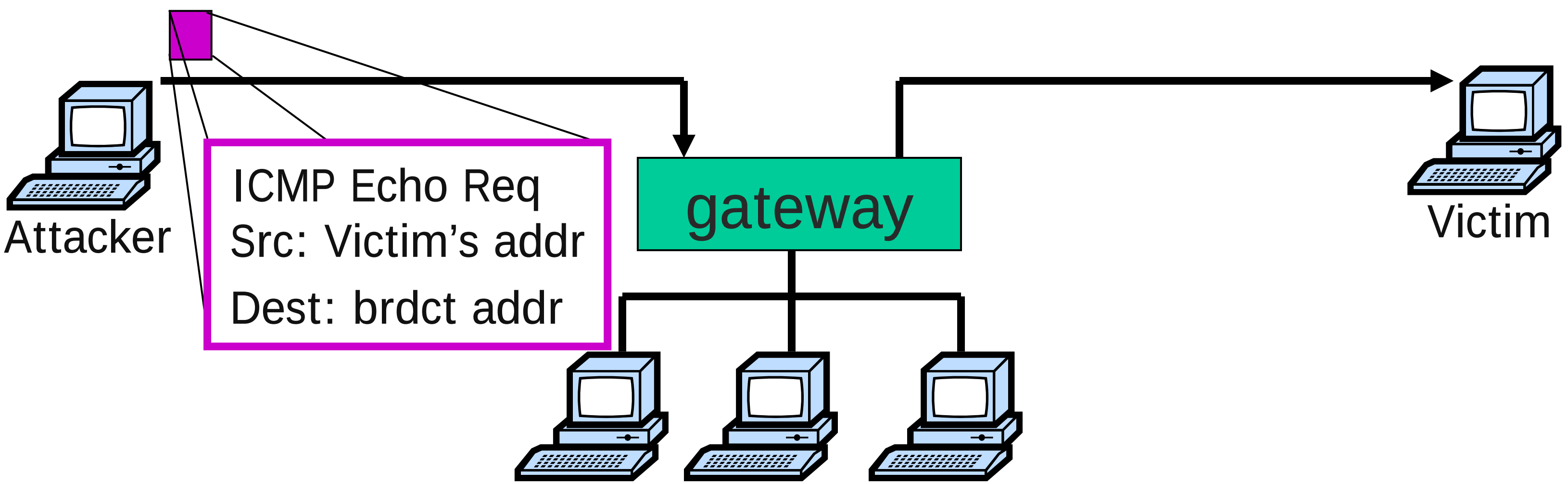
Attacker's Goals

- Hide
- Maximize damage

These goals are essential to understand what makes an attack effective and how to counter attacks

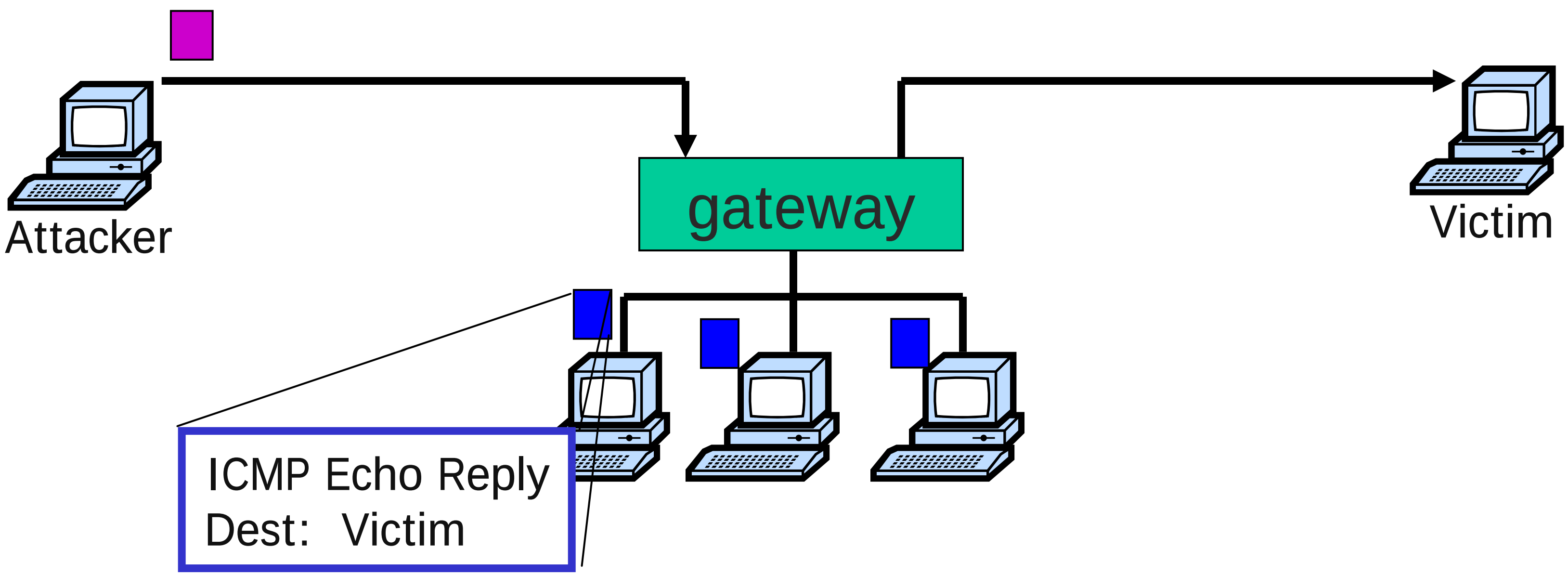
Attacker Wants to Hide

- Spoof the source (IP address, email account, ...)
- Indirection
  - Reflector attacks: E.g., Smurf Attack

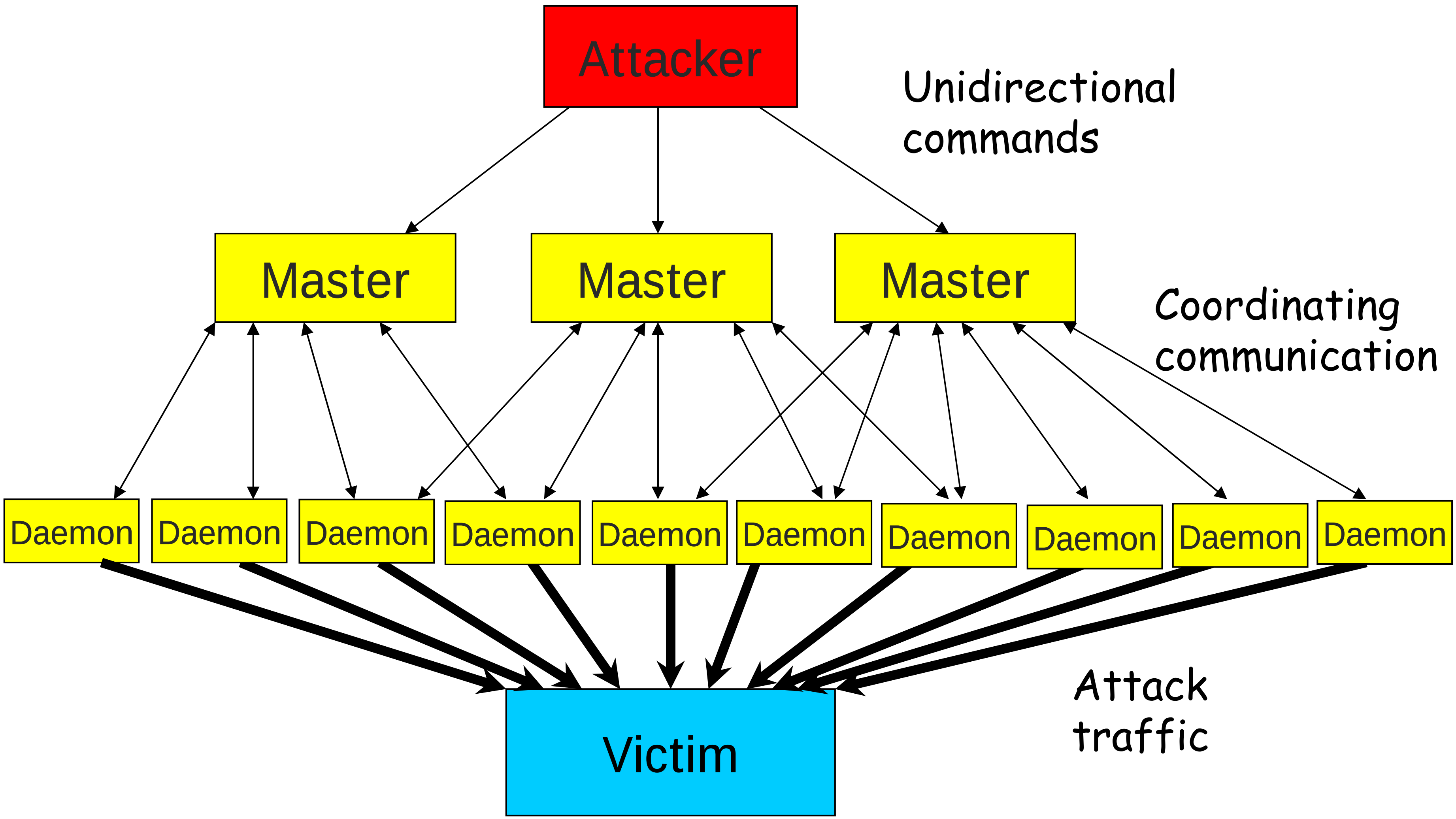


Attacker Wants to Hide

- Spoof the source (IP address, email account, ...)
- Indirection
  - Reflector attacks: E.g., Smurf Attack



Increase Damage → Go Fully Distributed → Use a Botnet



Some Distributed Denial of Service (DDoS) Tools

- Many public tools for flooding a victim with unwanted traffic
- Trin00 (Trinoo)
  - Client ported to Windows
- TFN – Tribe Flood Network
  - TFN2K – Updated for 2000
- Stacheldraht
  - German for "Barbed Wire"

Trin00

- a.k.a. "The Distributed DoS Project"
- Strengths
  - Password protected options, encrypted daemon list
    - Startup, remote control, and kill
  - Attacker talks to client using tcp
  - Master and daemons use udp
- Weakness
  - All messages (commands) sent in clear. Easy to fingerprint if network is infected

Trinoo Transcript

```
Connection to port (default 27665/tcp)

attacker$ telnet 10.0.0.1 27665
Trying 10.0.0.1
Connected to 10.0.0.1
Escape character is '^]'.
Kwijibo
Connection closed by foreign host. . . .


attacker$ telnet 10.0.0.1 27665
Trying 10.0.0.1
Connected to 10.0.0.1
Escape character is '^]'.
Betaalmostdone
trinoo v1.07d2+f3+c..[rpm8d/cb4Sx/]

trinoo>
```

TrinOO Commands

- **dos <IP> - command to initiate a DoS against the targeted <IP> address**
- **mdos <IP1:IP2:IP3> - sends command to attack three IP addresses, sequentially**
- **die – shut down the master**
- **mdie <password> - if correct password specified, packet is sent out to all daemon nodes to shutdown**
- **mping – ping sent to all nodes in the deamon list**
- **killdead – delete deamon nodes from list that didn't reply to ping**
- **bcast – gives a list of all active daemons**
- **mstop – Attempts to stop an active DoS attack. Never implemented by the author(s), but the command is there**

Bots Stories

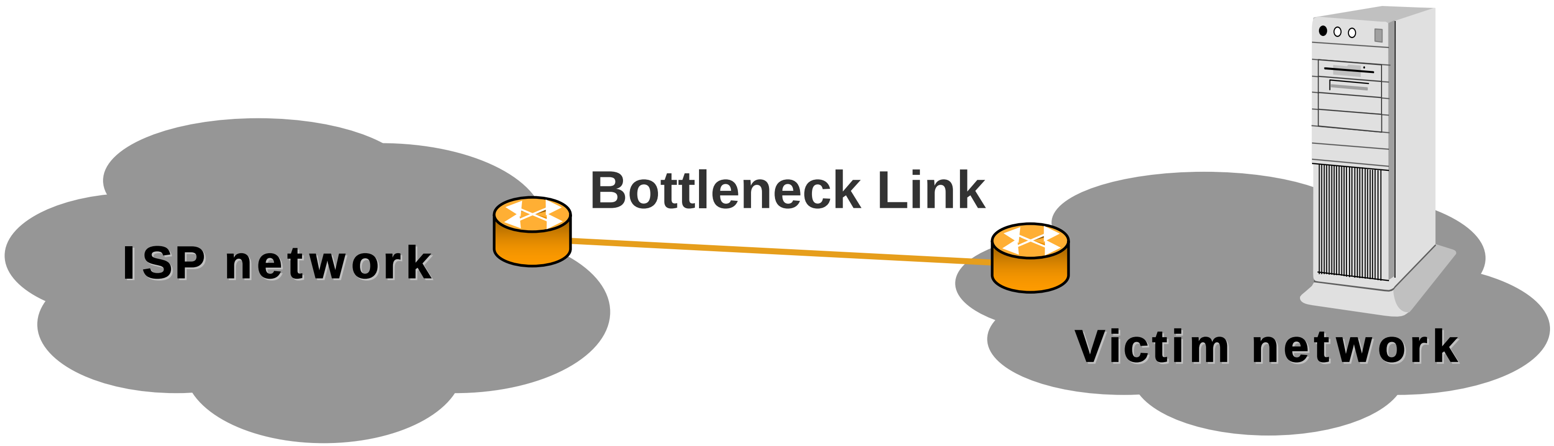
- Every day 30,000 new machines become zombies/bots
- Bots of 20,000+ machines are reported
- Bots are rented by the hour
- Bots are used for a variety of attacks, DDoS, Spam, as web servers which serve illegal content,…

Attacks

Attacks on Bandwidth

- Brute force attack
- Attacker sends traffic to consume link bandwidth
- What kind of packets?
  - ICMP Echo (e.g., TFN); UDP data (e.g., Trinoo, TFN); Junk TCP data or Ack packets (Stacheldraht v2.666, mstream); TCP SYN packets (TFN, Stacheldraht)

Defending against bandwidth attacks is hard

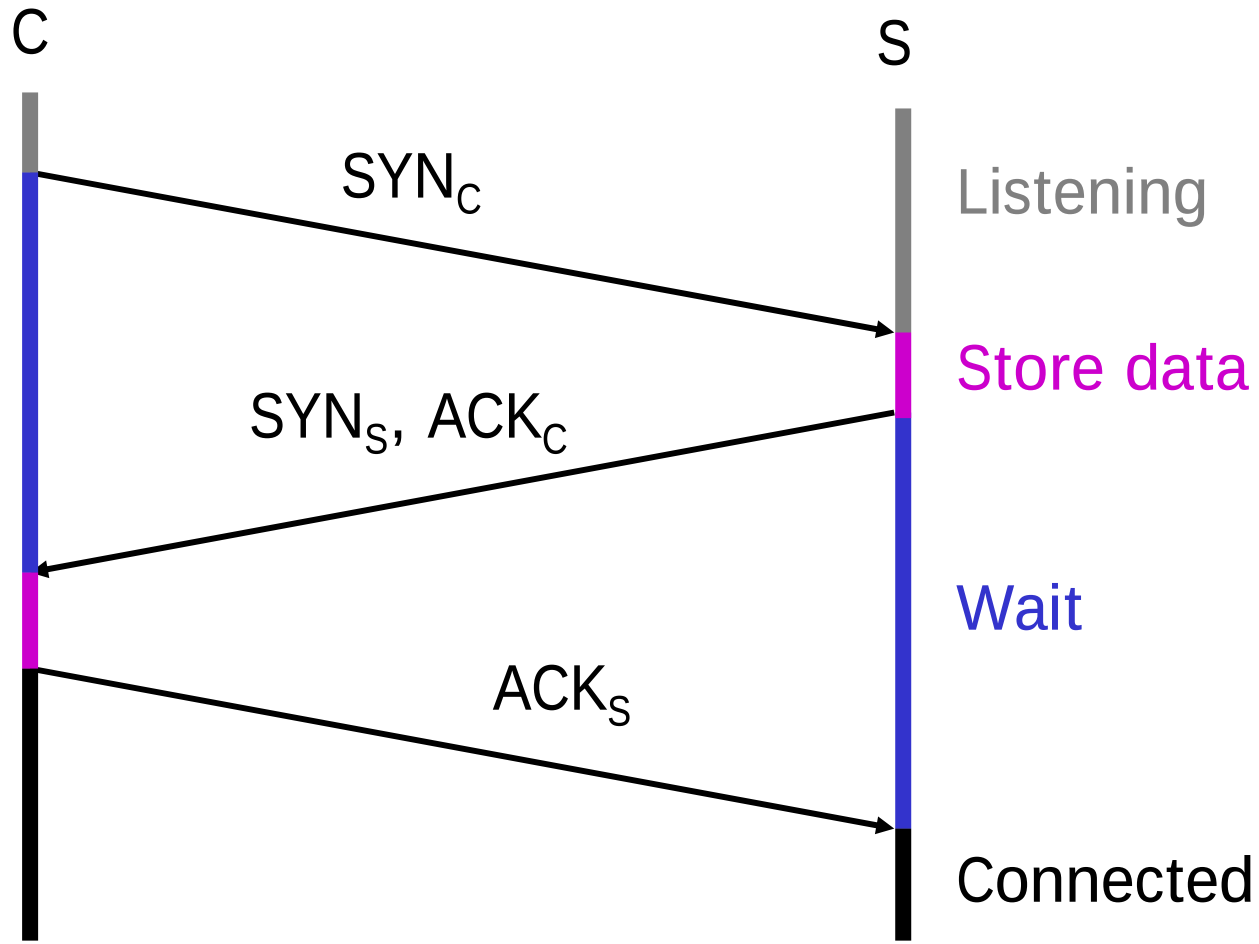


- Should drop packets before the bottleneck, i.e., at ISP
- But
  - ISPs are not willing to deploy complex filters for each client
  - ISPs have no strong incentive; they charge clients for the traffic
- Big companies defend themselves by using very high bandwidth access links

Attacks on TCP

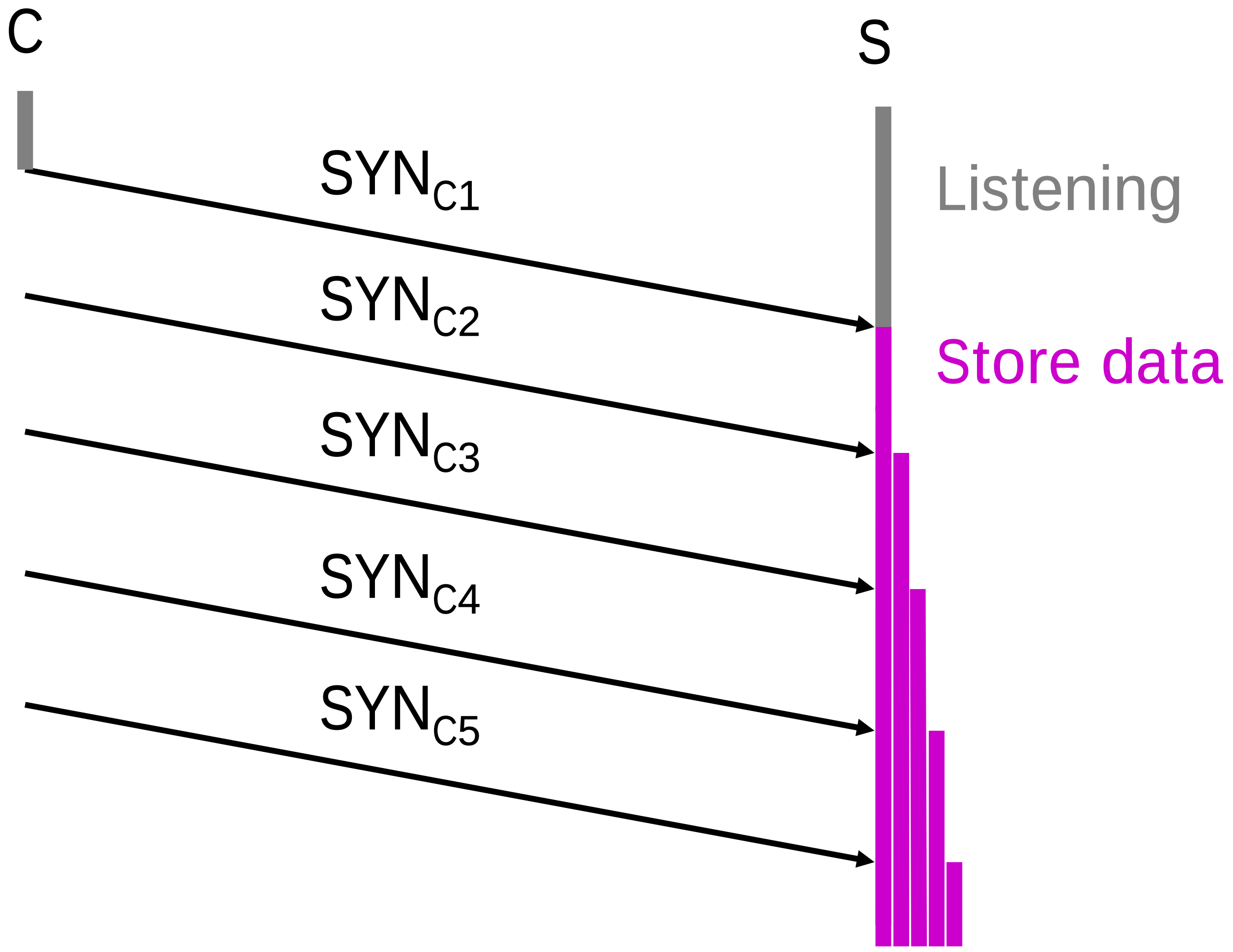
<u>TCP DoS Attacks:</u>

TCP SYN Flood



TCP DoS Attacks:

TCP SYN Flood



TCP DoS Attacks:

TCP SYN Flood

- Usually targets connection memory → Too many half-open connections

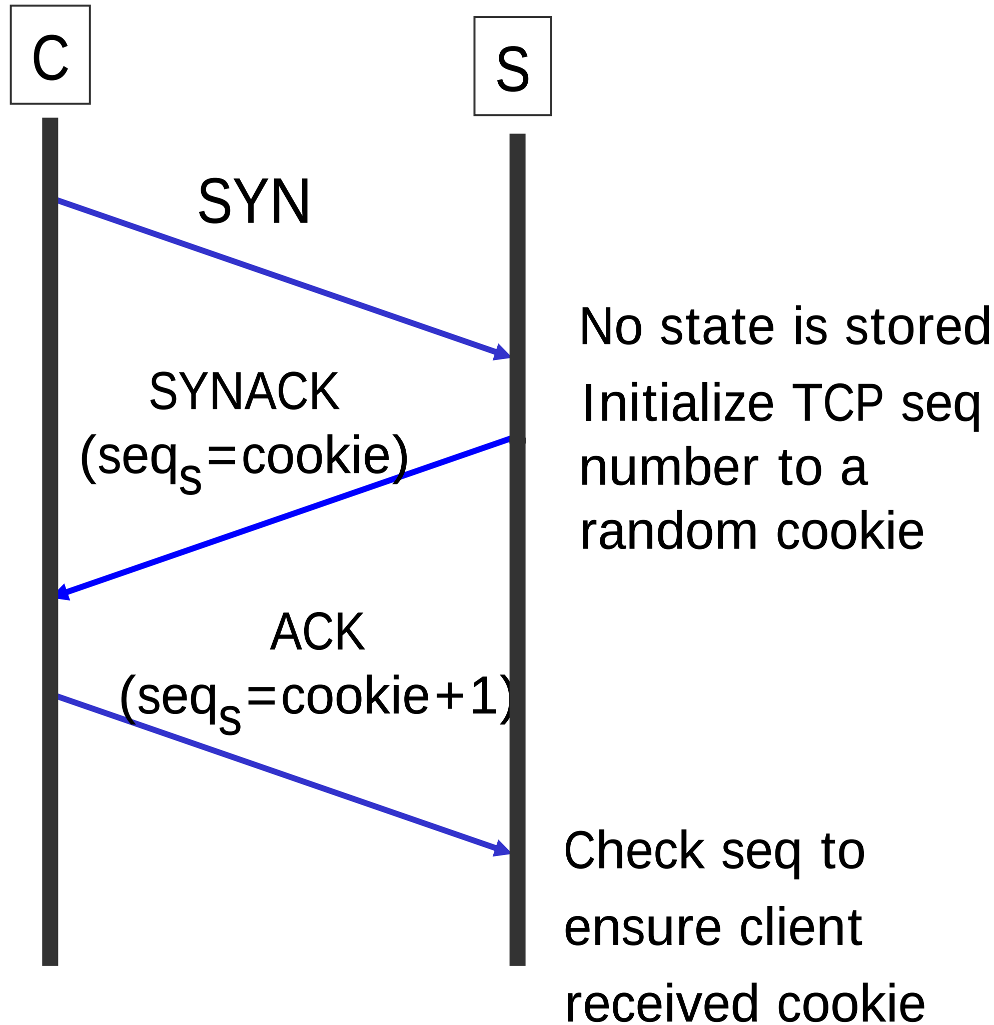
- Potential victim is any TCP-based server such as a Web server, FTP server, or mail server

- To check for SYN flood attacks
  - Run `netstat -s |grep "listenqueue overflows“` and check whether many connections are in "SYN_RECEIVED"

- How can the server deal with it?
  - Server times out half-open connection
  - SYN cookies and SYN caches prevent spoofed IP attacks

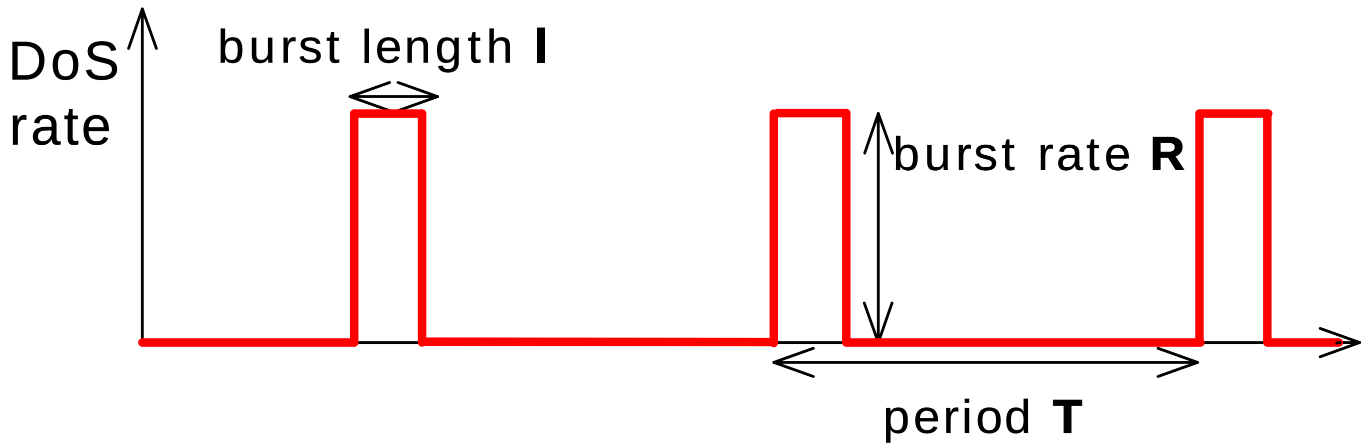
SYN Cookie

- Ensures IP address is not spoofed
- How? check that the client can receive a packet at the claimed source address



TCP DoS Attacks:

Low Rate TCP-Targeted Attacks

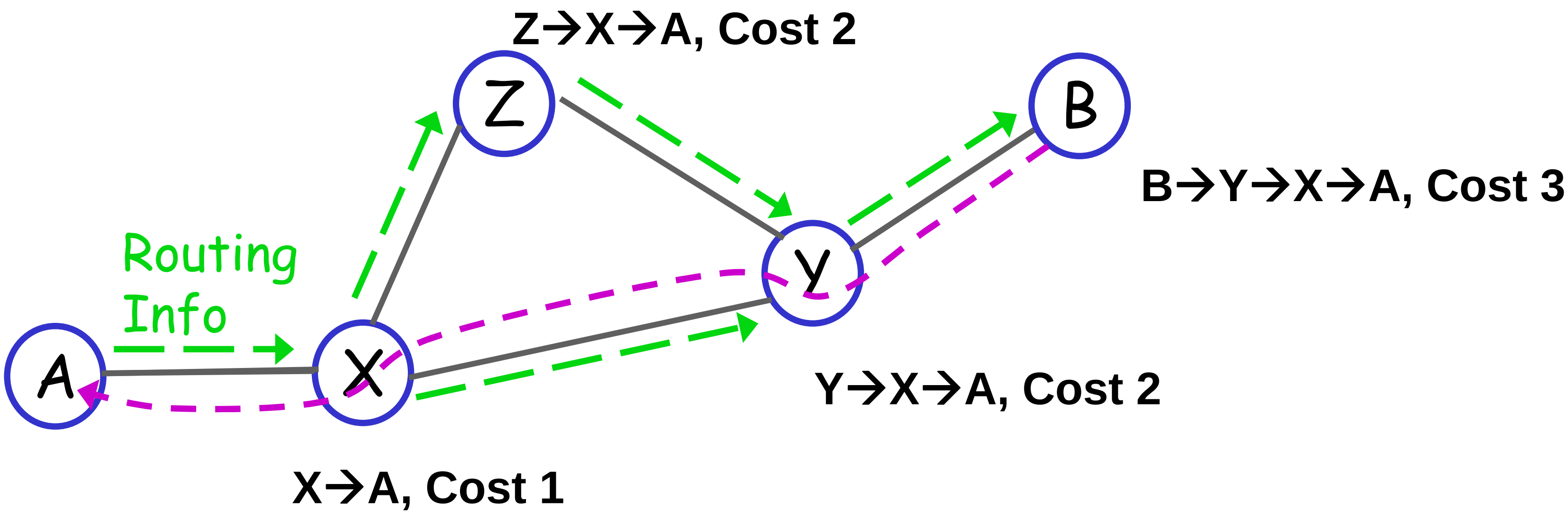


- Provoke a TCP to repeatedly enter retransmission timeout by sending a square-wave (l~RTT, T~minRTO)
- Hard to detect because of its low average bandwidth
- Randomizing TCP timeout helps but doesn't solve problem

Attacks on Routers

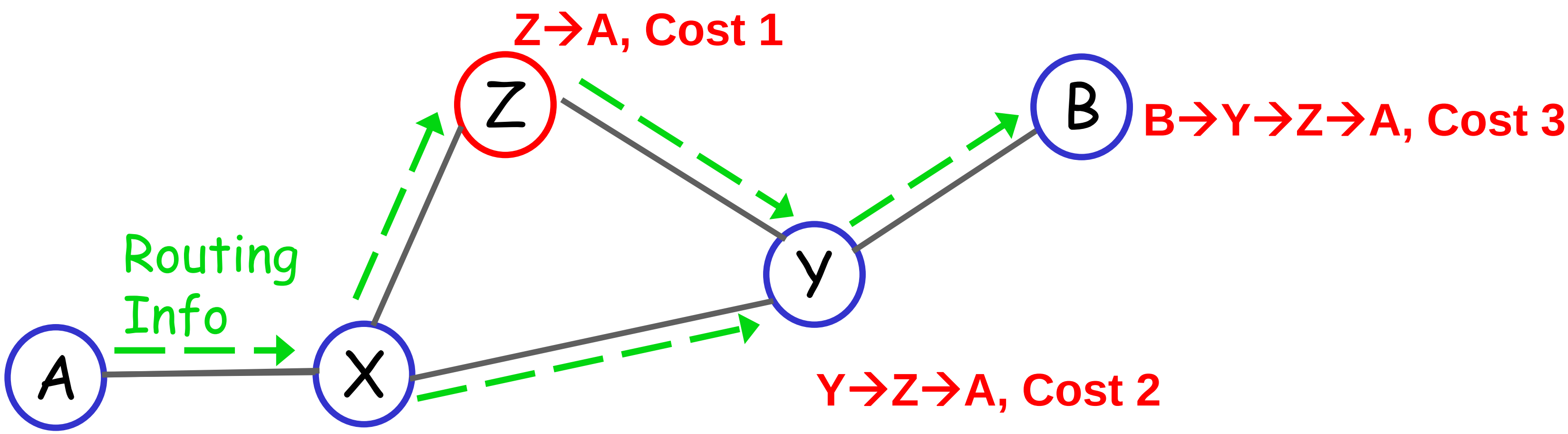
Attacks on Routers:

Routing Protocols



Attacks on Routers:

Attacks on Routing Table



- Attacker needs to get access to a router
- Attacks
  - Prefix hijacking by announcing a more desirable route
    - Z can lie about its route to A
  - Overload routers CPU by too many routing churns
  - Overload the routing table with too many routes
    - Causes router to run out of memory or CPU power for processing routes
    - E.g., AS7007

Countering Routing Table Attacks

- Authenticate routing adjacencies
- ISPs should filter routing advertisements from their customers
- Secure BGP [Kent et al]
  - Every ISP sign its advertisements creating a chain of accountability (e.g., X sends $\{ Z: \{Y\}_Z \}_X$
  - Too many signatures → too slow
    - With no authentication needs a few usec; MD5 ~100 usec; RSA ~1 sec

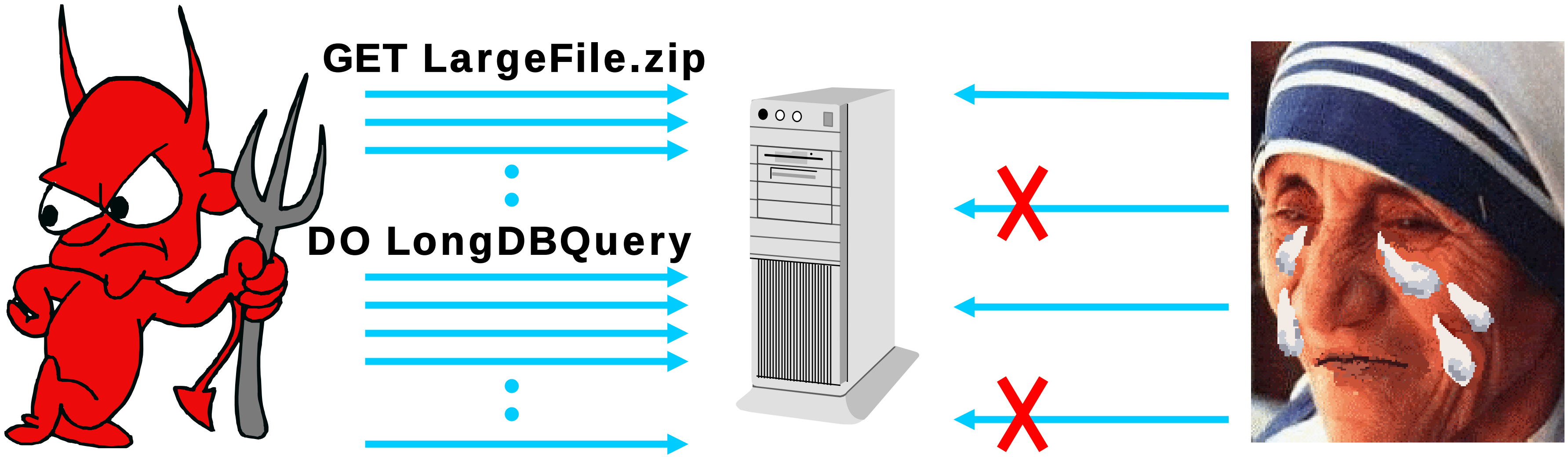
DoS Attacks on Web Servers

DoS Attacks on Web Servers

- Most known attacks
  - E.g., Yahoo, Amazon, …
  - Moore et al report over 12,000 attacks in 3-week, intentity as high as 600,000 pkts/s
- Recently taking the form of Cyber Mafia
  - Pay us $50,000 to protect you from attacks similar to the one on last Tuesday
- Becoming more distributed
  - Less spoofing of IP addresses
- Attack types
  - Attacks on TCP or Link bandwidth can be used against a Web server
  - Attacks on higher level protocols like HTTP

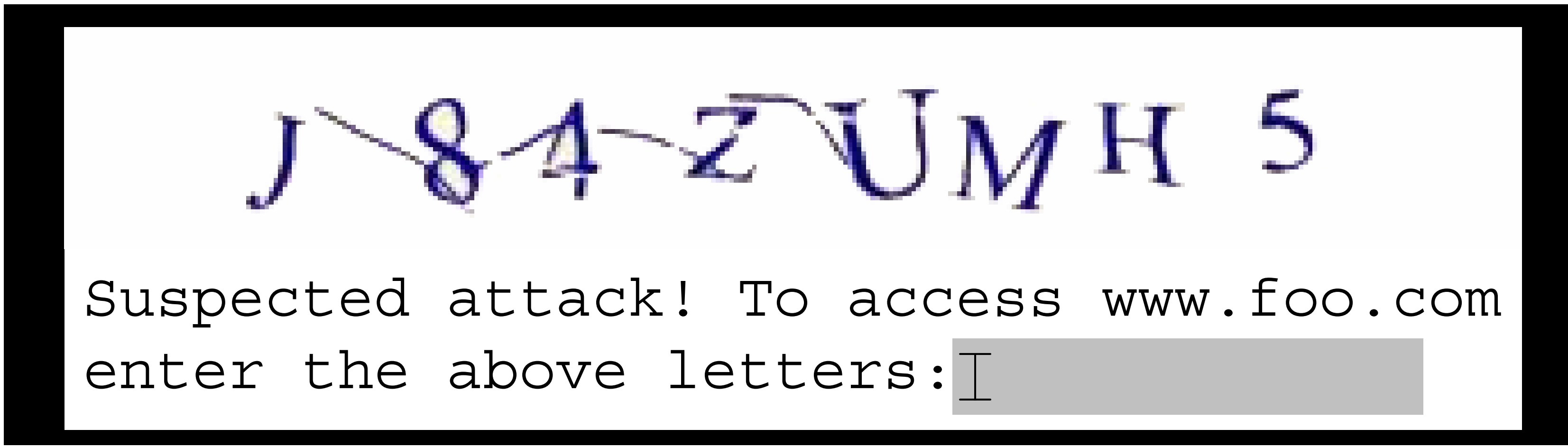
DoS Attacks on Servers:

Attacks that Mimic Legitimate Traffic



- Attacker compromises many machines causing them to flood victim with HTTP requests (e.g., MyDoom worm)
- Attacked resources
  - DB and Disk bandwidth
  - Socket buffers, processes, …
  - Dynamic content, password checking, etc.
- Hard to detect; attack traffic is indistinguishable from legitimate traffic

Proposals for Graphical Solutions



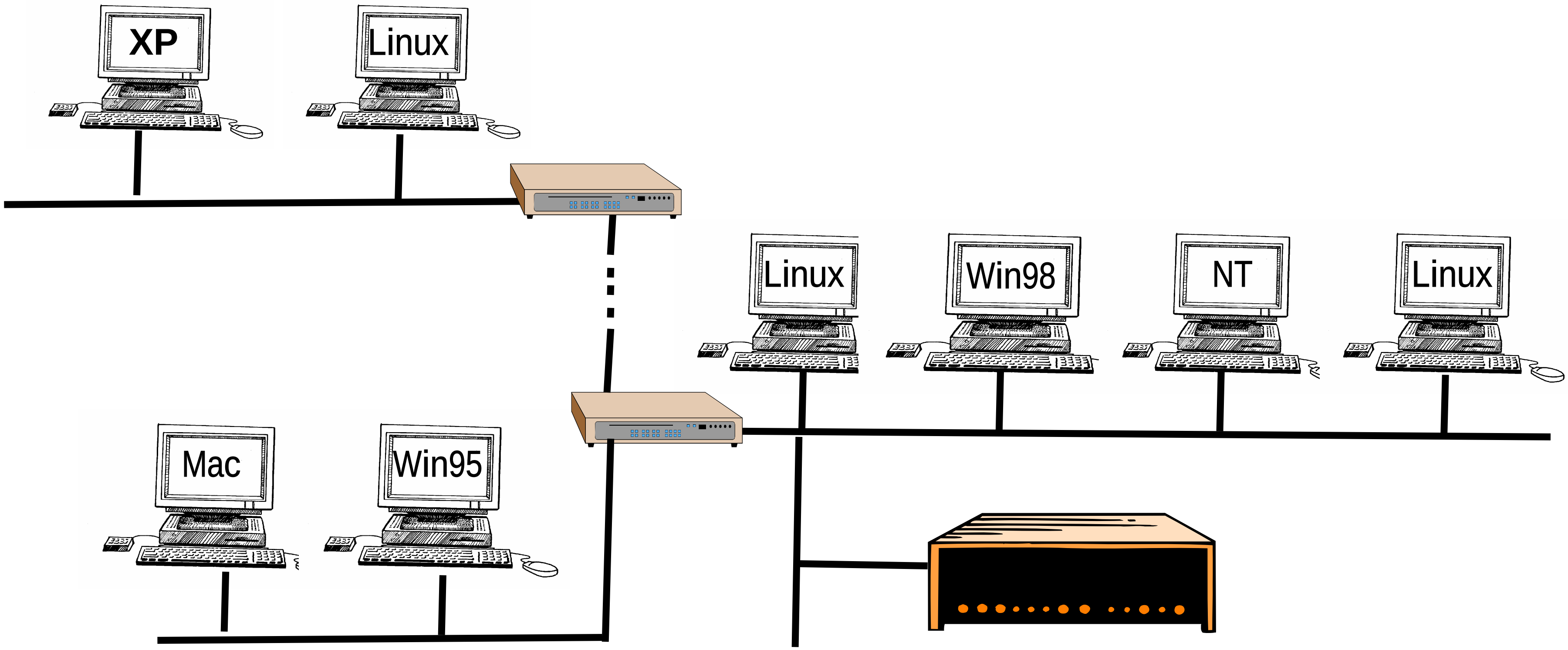
- Not that simple:
  - Should send test and check answer without allowing the unauthenticated clients access to server resources, including TCP sockets. Otherwise attack is accomplished.
  - Some people can't or don't want to answer graphical tests but are legitimate users

Detection

Detection Issues

- Detecting What?
  - Detecting the offending packets
  - Some attack characteristics (e.g., how many zombies)
  - The occurrence of an attack
- Offline vs. realtime
  - Realtime detection may help in throttling the attack while forensics might help in suing the attacker
- Detection cost
  - Can attacker mount an attack on the detection mechanism? How would that affect the protected system?

Network Intrusion Detection



- NIDS box monitors traffic entering and leaving your network
- In contrast to firewalls, NIDS are passive

Approaches to Intrusion Detection

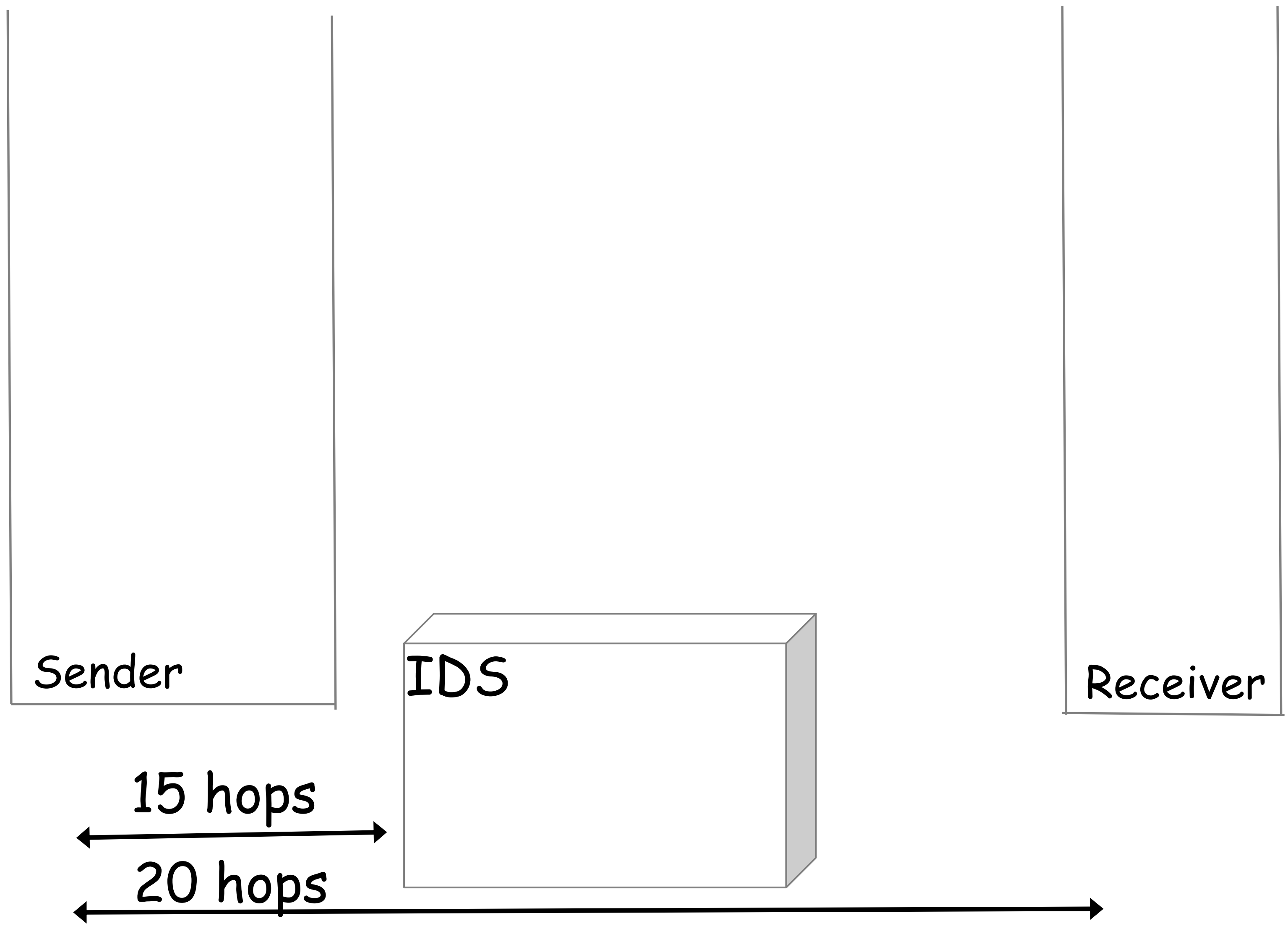
1. Signature Based: Keeps a DB of known attack signatures and matches traffic against DB (e.g., Bro, Snort)
   - Pros
     - Easy to understand the outcome
     - More accurate in detecting known attacks
   - Cons
     - Can't discover new attacks
2. Anomaly Based: Matches traffic against a model of normal traffic and flags abnormalities (e.g., EMERALD)
   - Pros
     - Can deal with new attacks
   - Cons
     - Modeling normal. it is hard to describe what is normal
     - Limits new applications
     - Less accurate detection of known attacks
3. Hybrid: Matches against DB of known attacks. If no match, it checks for anomaly

Evasion Problem in NIDS

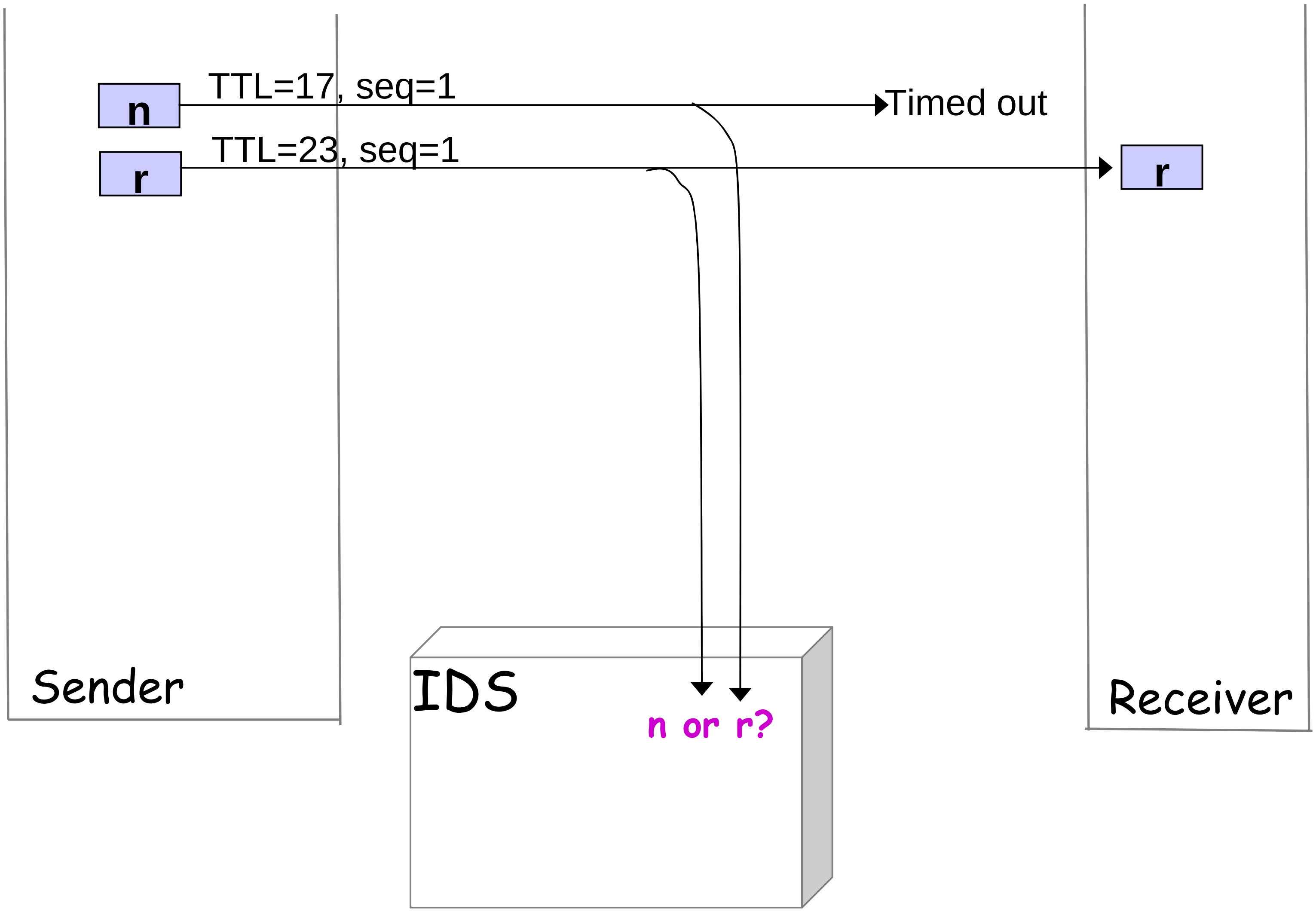
- Consider scanning traffic for a particular string ("`USER root`")
- Easiest: scan for the text in each packet
  - No good: text might be split across multiple packets
- Okay, remember text from previous packet
  - No good: out-of-order delivery
- Okay, fully reassemble byte stream
  - Costs **state** ….
  - …. and still evadable

Source: Vern Paxson

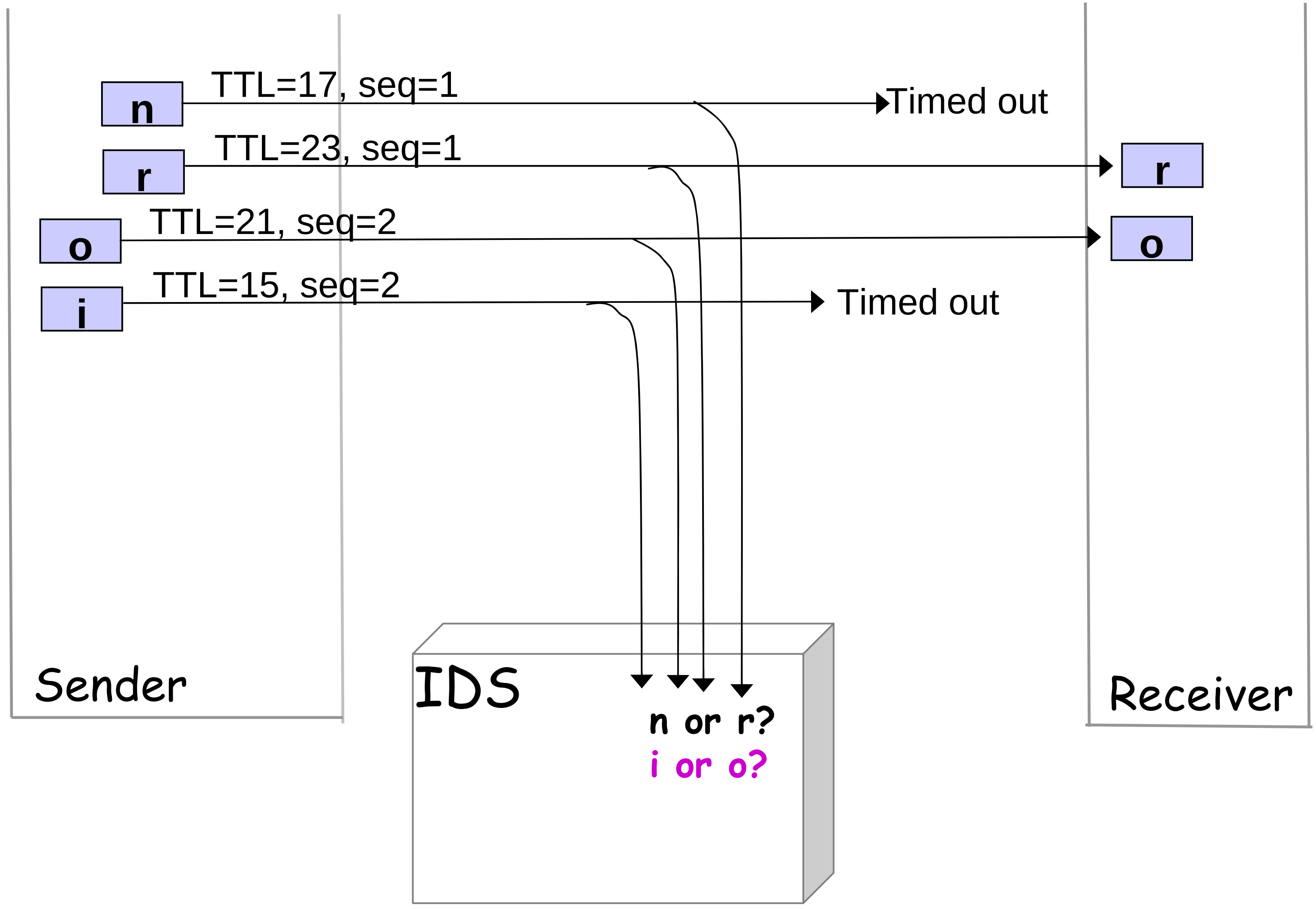
Evading Detection Via Ambiguous TCP Retransmission



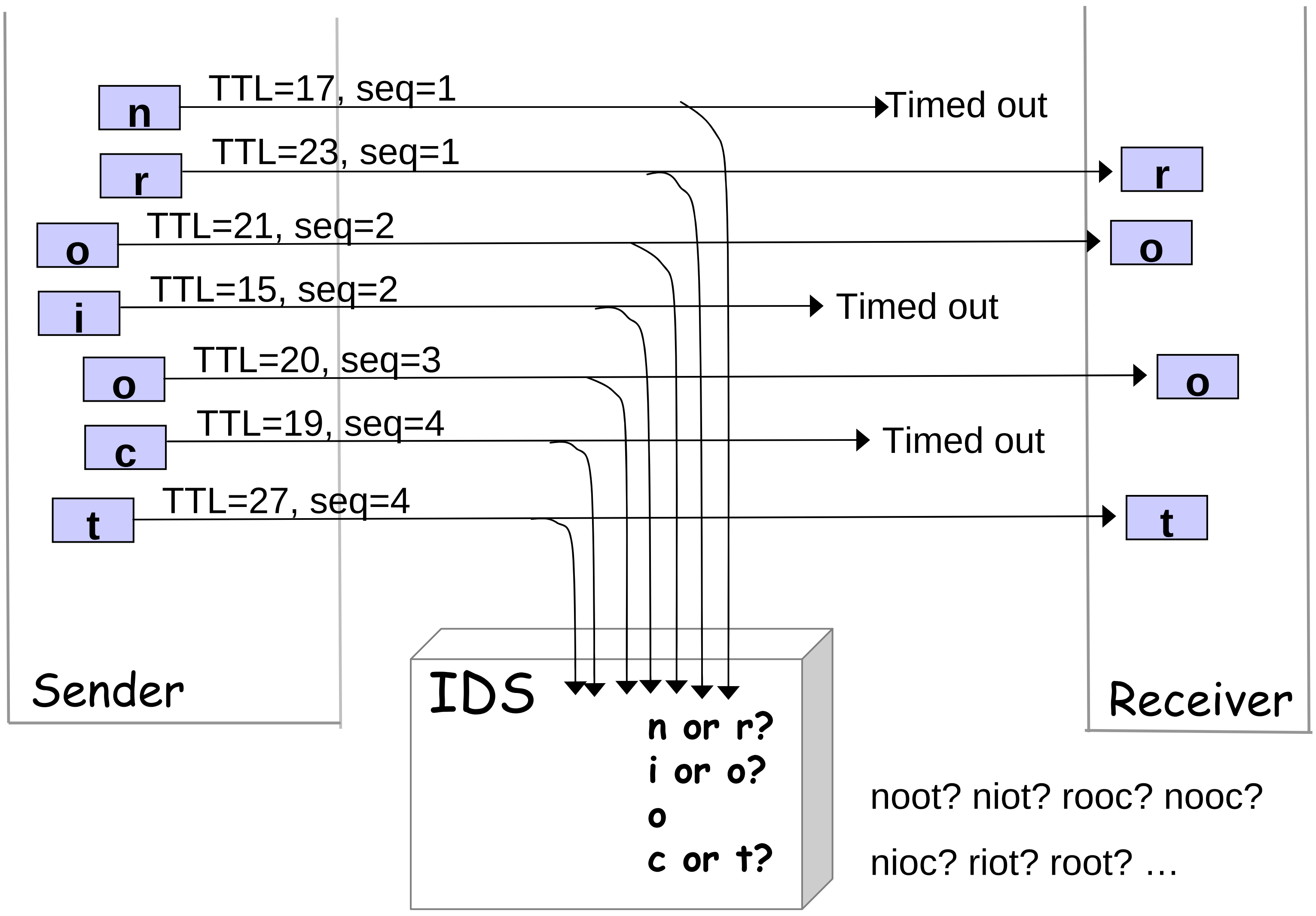
Evading Detection Via Ambiguous TCP Retransmission



Evading Detection Via Ambiguous TCP Retransmission



Evading Detection Via Ambiguous TCP Retransmission



noot? niot? rooc? nooc?

nioc? riot? root? …

Bypassing NIDS

- Evasion
- Insertion
- DoS it
- Hack it
- Cause many false alarms until admin stops paying attention

Examples of Anomaly Detection

- Detecting Large Bandwidth Consumers
- MULTOPs
- Distinguishing DDoS from flash-crowd

Detecting Malicious TCP Flows

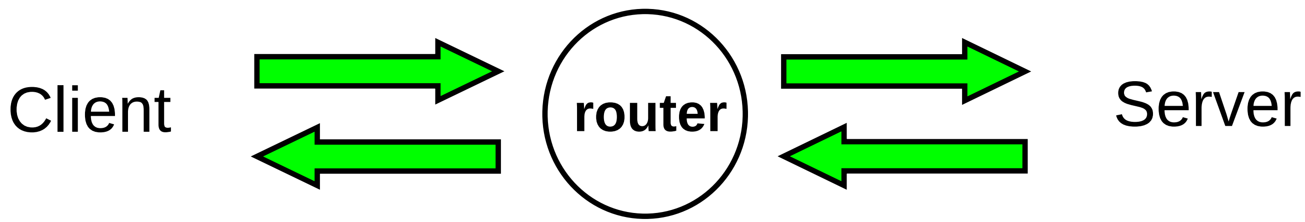
- TCP throughput is a function of its drop rate

$$Thru = \frac{1.2}{RTT * \sqrt{drop\_rate}}$$

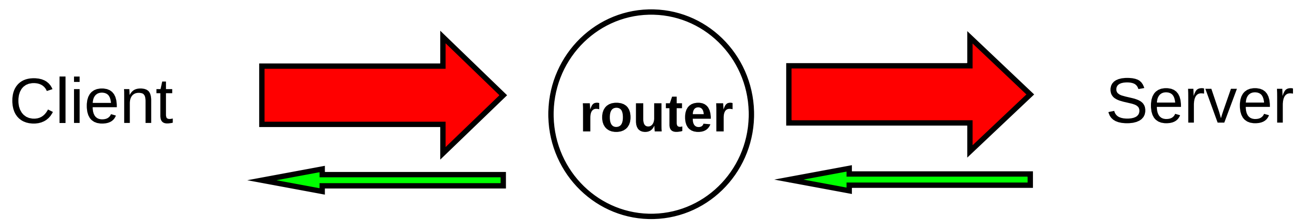
- Router monitors the rate of each TCP flow and compares it against the above equation
- Make it more scalable by using statistical monitoring at routers to find unfriendly flows

MULTOPS
protects web servers against BW attacks

HTTP Traffic is mostly from server to client



Normal: proportional packet rates



Attack: disproportional packet rates

Drop packets from sources sending disproportionate flows

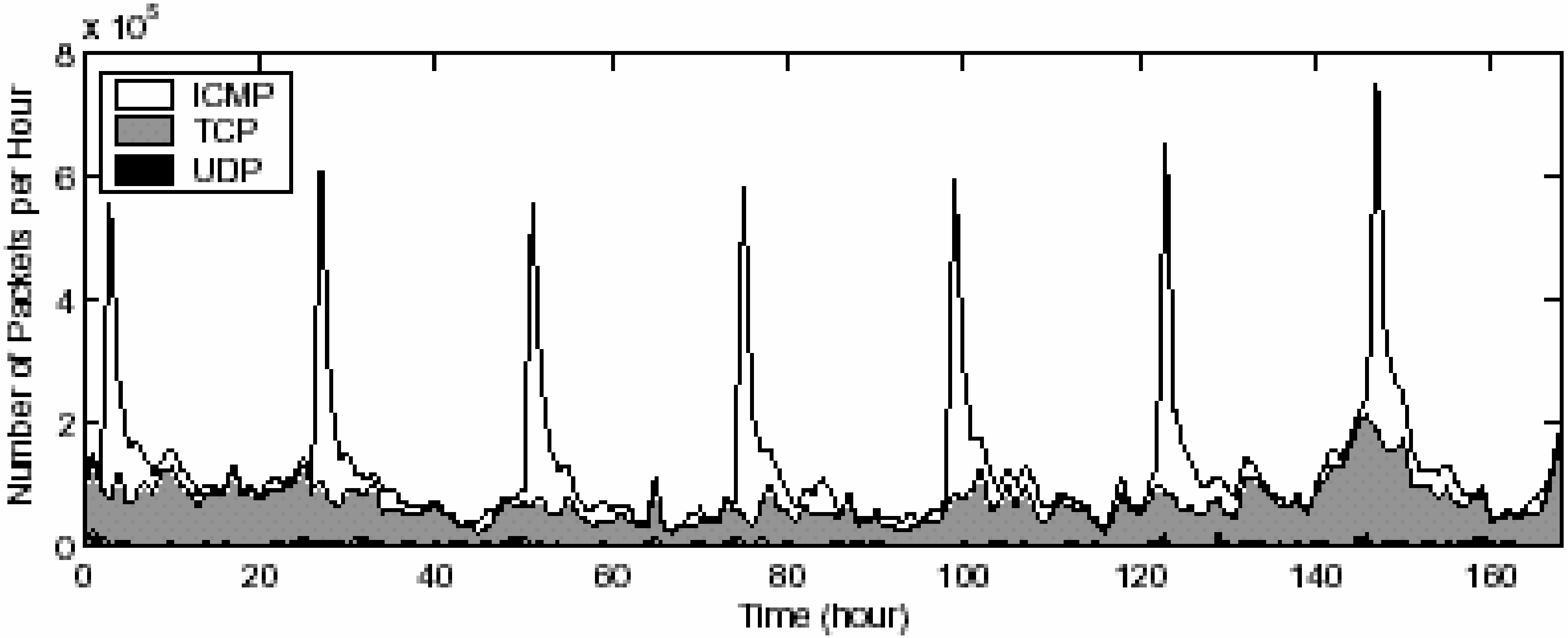
Distinguish DDoS Attacks from Flash Crowd

- Jung et al. identify whether overload is created by flash crowd or DDoS
- Idea: Prefixes of client addresses in DDoS attacks are randomly distributed, whereas in a flash crowd they are closer to the prefix distribution of the server's usual traffic

Network Telescopes

- Detect the occurrence of large scale abusive activities
- Idea: monitor an unused cross-section of Internet address space. Packets received at these unused addresses are signs of attacks
  - “Backscatter” from DoS (attacker spoof an address from monitored space causing the victim to reply to the monitor)
  - Attackers probing blindly
  - Random scanning from worms
- If you monitor 1/n the IP address space, by the time you observe the abusive activity it has affected about $2^{32}/n$ Internet hosts

Hourly Background Radiation Seen at a 2,560-Address Network Telescope



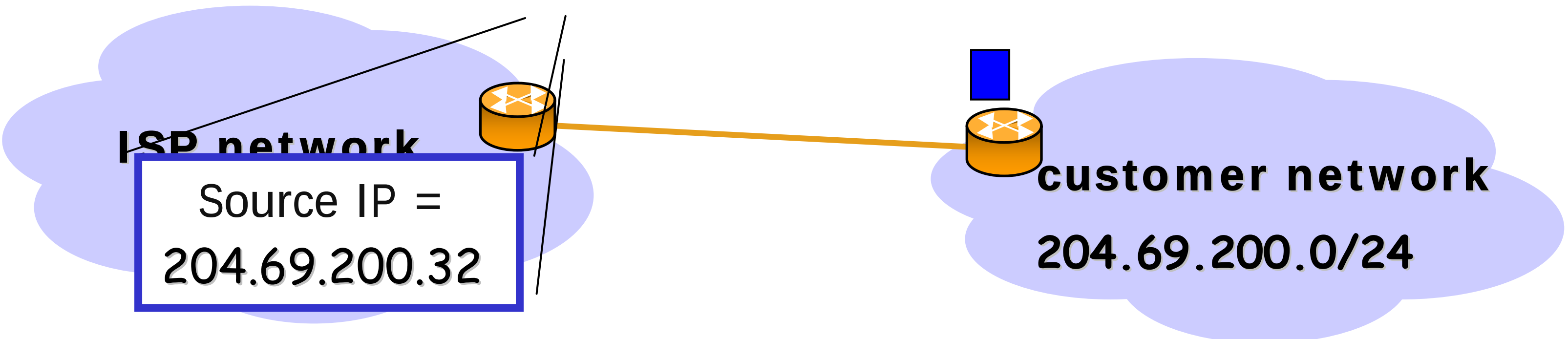
Source: Vern Paxson

Authentication & Establishing Identity

Establish Identity

- Find the source of the offending traffic
- Important for blacklisting, imposing legal/social charges, fix the zombies, …

Ingress Filtering

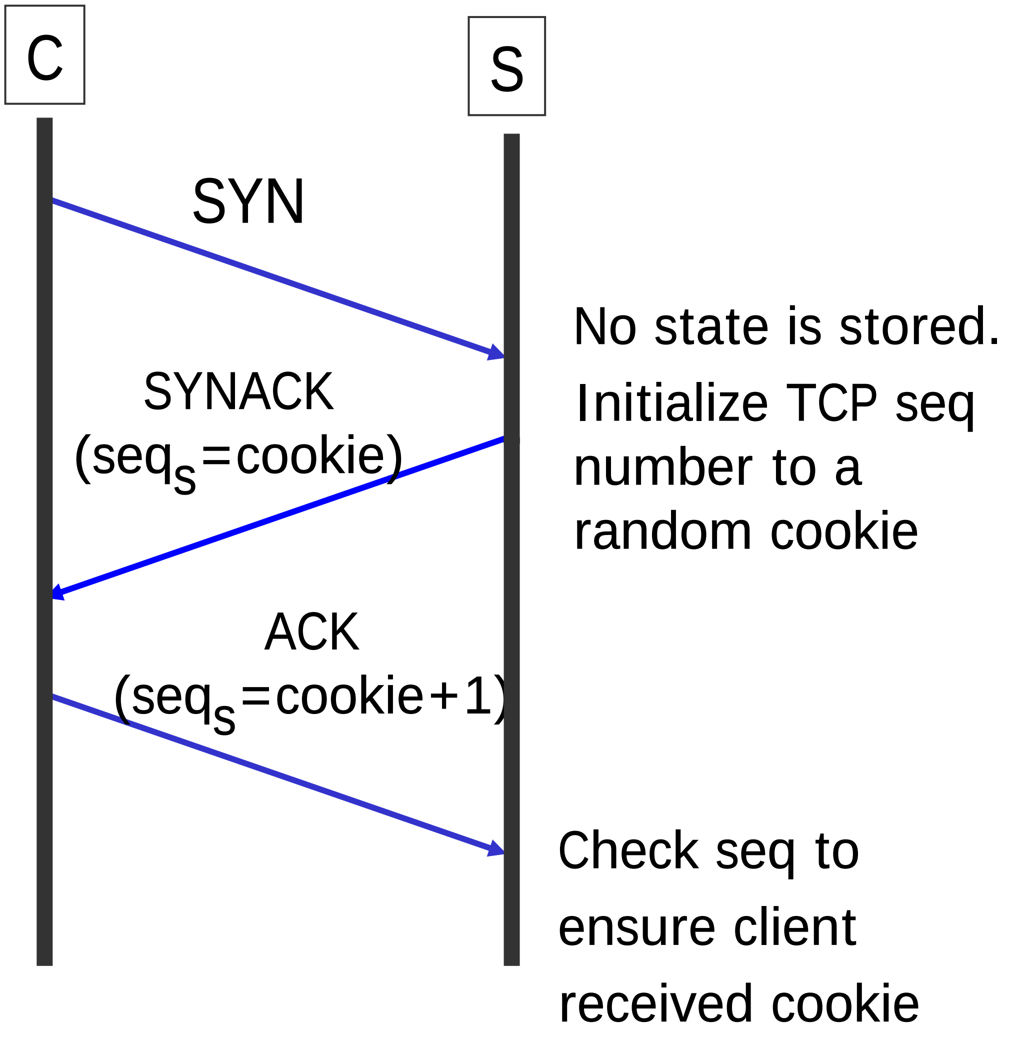


- An ISP checks that all packets from a customer network use a source address from the customer's allocated address space [RFC 2827]
- Also, the customer checks that all packets leaving its network have the correct source prefix

Methods for authenticating a source:

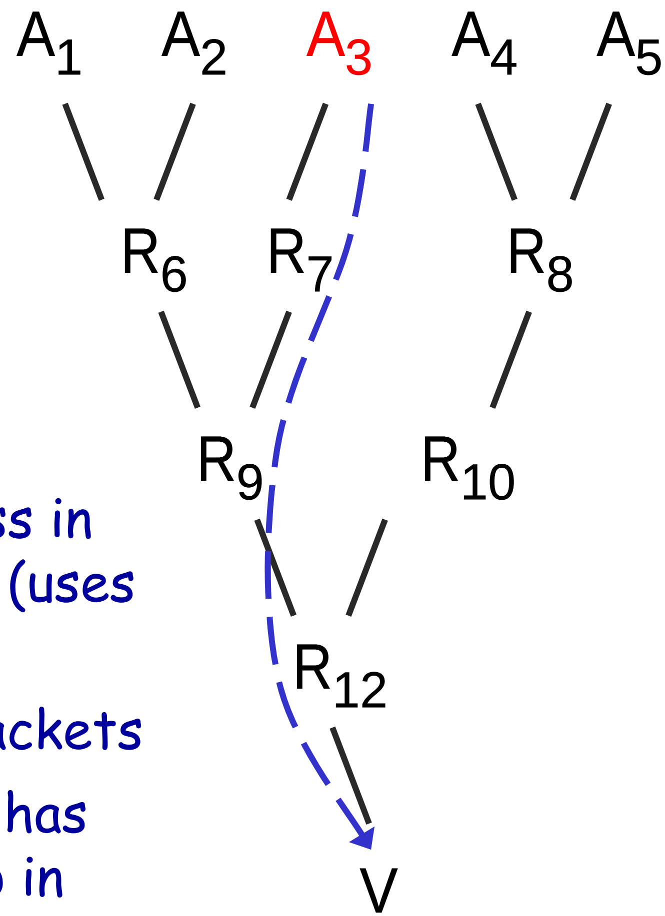
Different forms of pinning a route

- Circuit switching
- Virtual Circuit
- SYN Cookie



IP Traceback

- Relies on routers' help in detecting the attack path
  - Assume you trust routers
- Probabilistic traceback:
  - Every router writes its IP address in the packet with some probability (uses fragment field in IP header)
  - Victim reconstructs path from packets
  - Router at distance d from victim has probability $p(1-p)^{d-1}$ of showing up in marked packets



Authorization

- Who are the legitimate senders?
  - Private services → legitimate users have known IP addresses, known passwords, …
    - E.g., authentication of routing adjacencies
  - Public services → hard, don't know legitimate users
    - E.g., Google, Amazon, …
    - Should ask what makes a certain access pattern legitimate
      - Human User → graphical test, …
      - Reasonable number of http requests per IP address?
      - No weird connection behavior (keeping half-open connections for long time)
- Problems with checking authorization
  - Compromised machines may expose passwords and login info

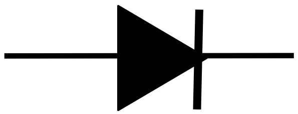
Cost of Checking Identity

- Cost << service/attack cost; Otherwise it is not worth it
  - Costly authentication schemes are prone to DoS attacks
    - E.g., attack on password authentication
  - Also costly authentication mechanisms slow down the service
    - secure BGP slows down routing making it hard to deploy

Filtering & Throttling

Firewalls

- A barrier between us and them
  - Limits communication to the outside world, so that only a few machines are exposed to attacks
- Semantics

inside  outside

- Why?
  - Most hosts have security holes
  - Firewalls run less code and hence have fewer bugs
  - Firewalls can be professionally administered

Possible Firewall Actions

- Access control (a list of good addresses and bad addresses)
- Ingress/egress filtering
  - Packets coming in must have outside source
  - Packets leaving must have an inside source
- Rate limiting
  - Limit rate of ICMP packets and/or SYN packets

- All of these steps may interfere with legitimate traffic
- They don't help when attacks come from inside

NAT (Network Address Translator) as a Firewall

- NAT deals with shortage of IPv4 addresses
  - why there is a shortage?
    - 2^{32} addresses; Hierarchical assignment
- Main idea behind NAT
  - Not all addresses are used at the same time globally
  - Many communications are local → don't need global addresses.

How does NAT work?

- Assign the global address to the NAT box
- Assign local addresses to machines behind the NAT (e.g., 10.0.0.0/8)
- Locally, advertise the NAT as the router connecting the network to the rest of the world → packets destined to outside destinations are going to leave through the NAT
- When a local host sends a packet to an outside destination
  - NAT capture the packet and replaces it source address and port
  - NAT adds binding to its table (Local_IP-Local_Port →global_IP-Global-Port)
  - NAT sends packet
  - When ack comes form the destination, NAT checks its table to replace the global source address and port with the local ones
  - Nat also checks the filter that should be applied to the incoming packet

NAT Functionality