CS244 Winter 2011

Lecture 17

Network Security #2

1. SIFF: A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks
   [A. Yaar, A. Perrig, D. Song]

Martin Casado

Internet Design vs. Security

Internet Design vs. Security

- Destination routing
- Packet based (statistical multiplexing)
- Global addressing (IP addresses)
- Simple to join (as infrastructure)
- Power in end hosts (end-to-end arg)
- “Ad hoc” naming system

Internet Design vs. Security

- Destination routing
  - Keeps forwarding tables small
  - Simple to maintain forwarding tables
  - How do we know where packets are coming from?
    - Probably simple fix to spoofing, why isn't it in place?
- Packet based (statistical multiplexing)
- Global addressing (IP addresses)
- Simple to join (as infrastructure)
- Power in end hosts (end-to-end arg)
- "Ad hoc" naming system

Internet Design vs. Security

- Destination Routing
- Packet Based (statistical multiplexing)
  - Simple + Efficient
  - Difficult resource bound per-communication
    - How to keep someone from hogging? (remember, we can't rely on source addresses)
- Global Addressing (IP addresses)
- Simple to join (as infrastructure)
- Power in End Hosts (end-to-end arg)
- "Ad hoc" naming system

Internet Design vs. Security

- Destination routing
- Packet based (statistical multiplexing)
- Global Addressing (IP addresses)
  - Very democratic
  - Even people who don't necessarily want to be talked to ("every psychopath is your next door neighbor" – Dan Geer)
- Simple to join (as infrastructure)
- Power in end hosts (end-to-end arg)
- "Ad hoc" naming system

Internet Design vs. Security

- Destination routing
- Packet based (statistical multiplexing)
- Global addressing (IP addresses)
- Simple to join (as infrastructure)
  - Very democratic
  - Misbehaving routers can do very bad things
  - No model of trust between routers
- Power in End Hosts (end-to-end arg)
- “Ad hoc” naming system

Internet Design vs. Security

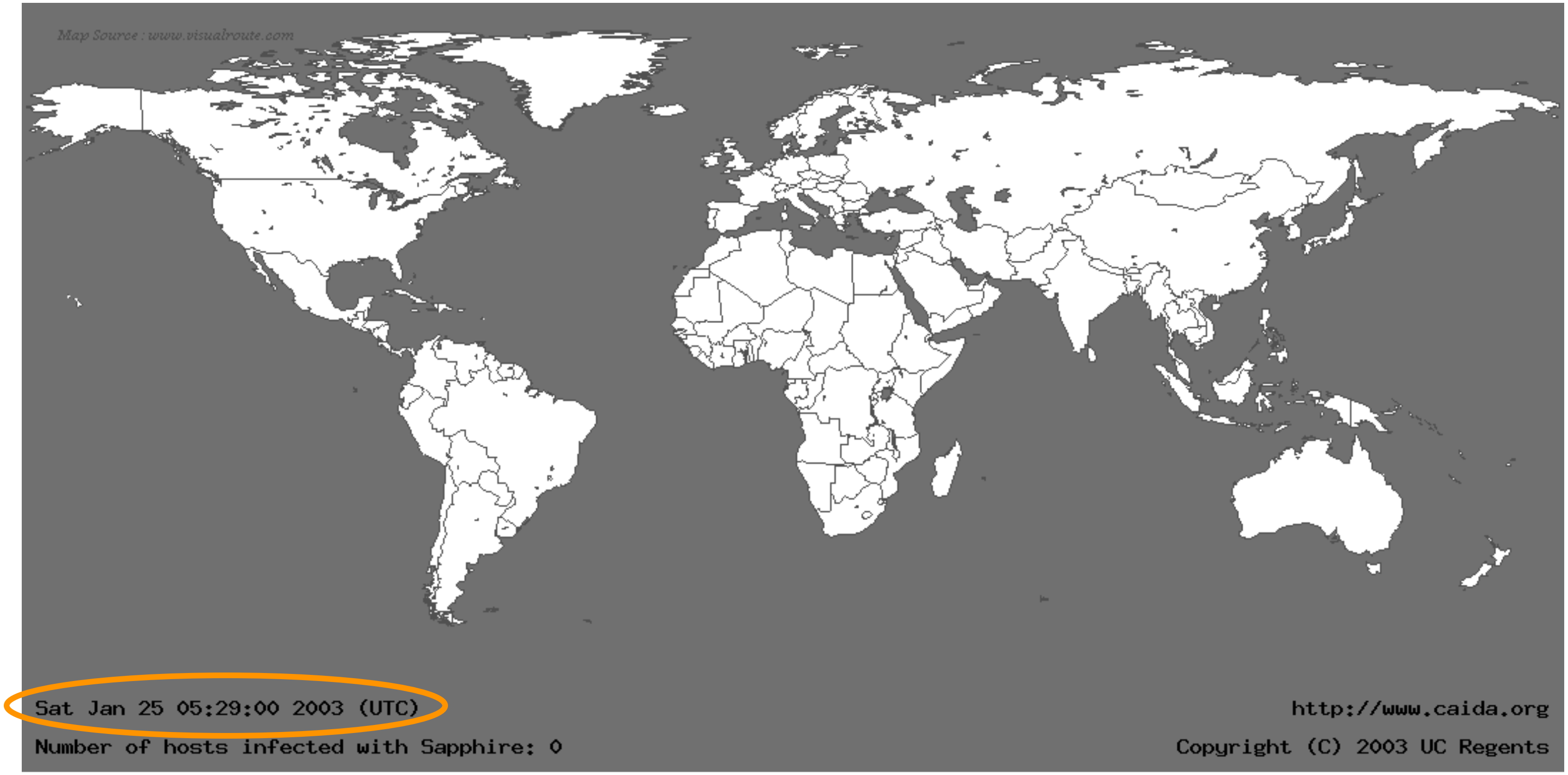
- Destination routing
- Packet based (statistical multiplexing)
- Global addressing (IP addresses)
- Simple to join (as infrastructure)
- Power in end-hosts (end-to-end arg)
  - Decouple hosts and infrastructure = innovation at the edge!
  - Giving power to least trusted actors
    - How to guarantee good behavior?
- “Ad hoc” naming system

Internet Design vs. Security

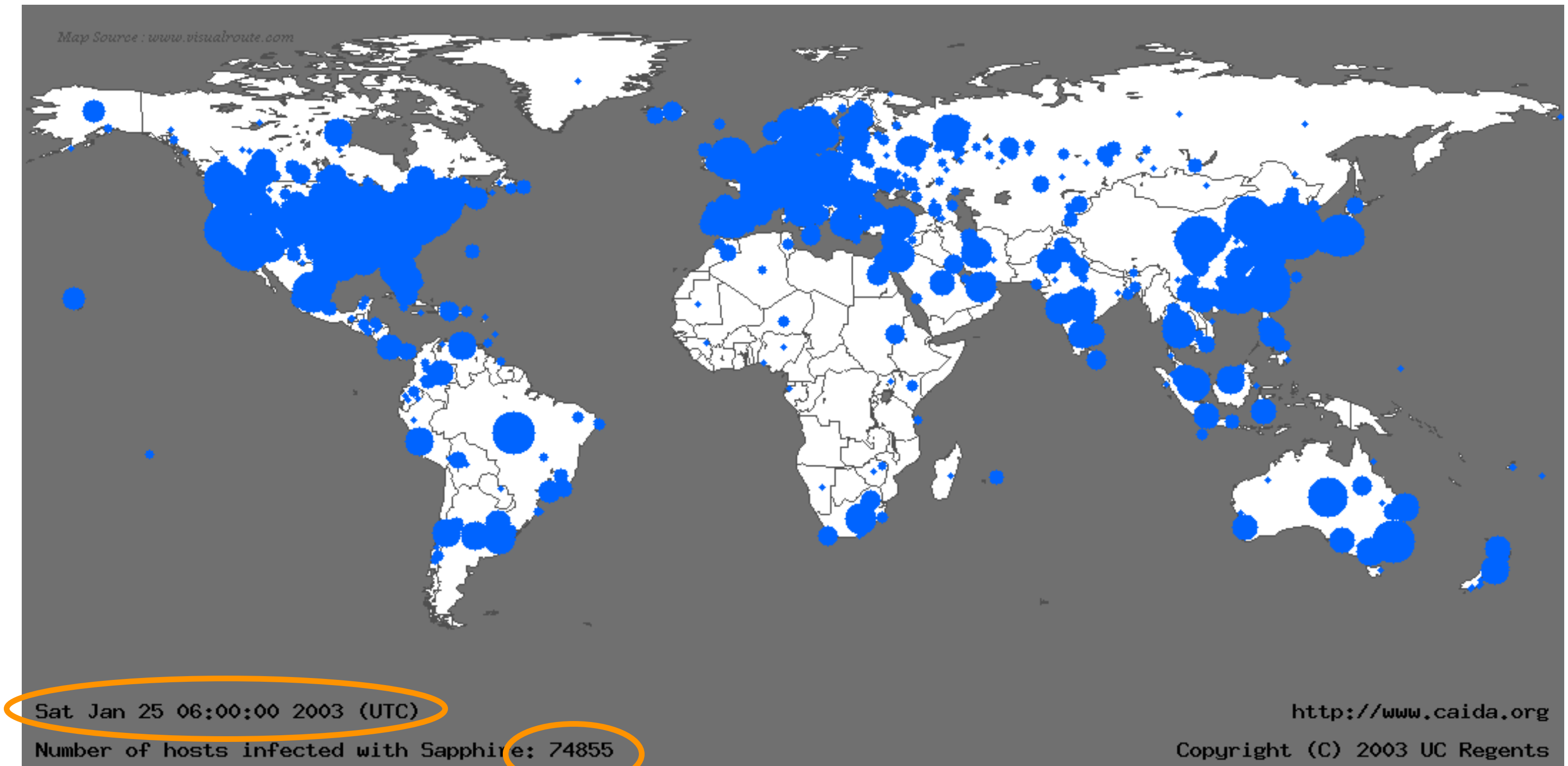
- – Packet Based (statistical multiplexing)
- – Destination Routing
- – Global Addressing (IP addresses)
- – Simple to join (as infrastructure)
- – Power in End Hosts (end-to-end arg)
- – "Ad hoc" naming system
  - Seems to work OK
  - Fate sharing w/ hierarchical system
  - Off route = more trusted elements

2 Anecdotes

Life Just Before Slammer



Life Just After Slammer



A Lesson in Economy

- Slammer exploited connectionless UDP service, rather than connection-oriented TCP.
- *Entire worm* fit in a single packet! (376 bytes)

  ⇒ When scanning, worm could "fire and forget".

  Stateless!
- Worm infected 75,000+ hosts in 10 minutes (despite broken random number generator).
  - At its peak, **doubled every 8.5 seconds**
- Progress limited by the Internet's *carrying capacity* (= 55 million scans/sec)

Impact

- First victim at 12:15am
- By 12:45, transcontinental links starting to fail
- 300,000 access points downed in Portugal
- All cell and Internet in Korea failed (27 million people)
- 5 root name servers were knocked offline
- 911 didn’t respond (Seattle)
- Flights canceled

Blue Security

- Anti-spam software Blue Frog
- Retaliation starting May 1, 2006
- Traffic black-holed by social engineering
- DDoS takes down original site
- Flooding disrupts operations of 5 top-tier hosting providers (including tucows)
- Blue security “folds” (May 15, 2006)
- Reportedly initiated by single attacker

DoS

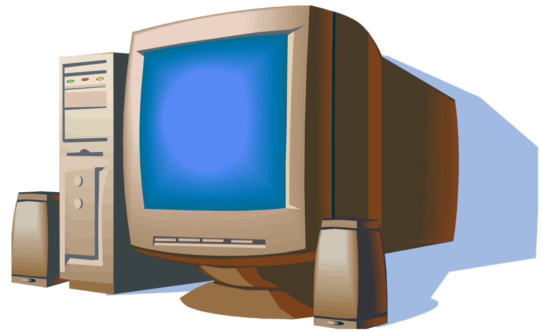
DoS: Via Resource Exhaustion

DoS: Via Resource Exhaustion

CPU

User-time

Uplink bandwidth



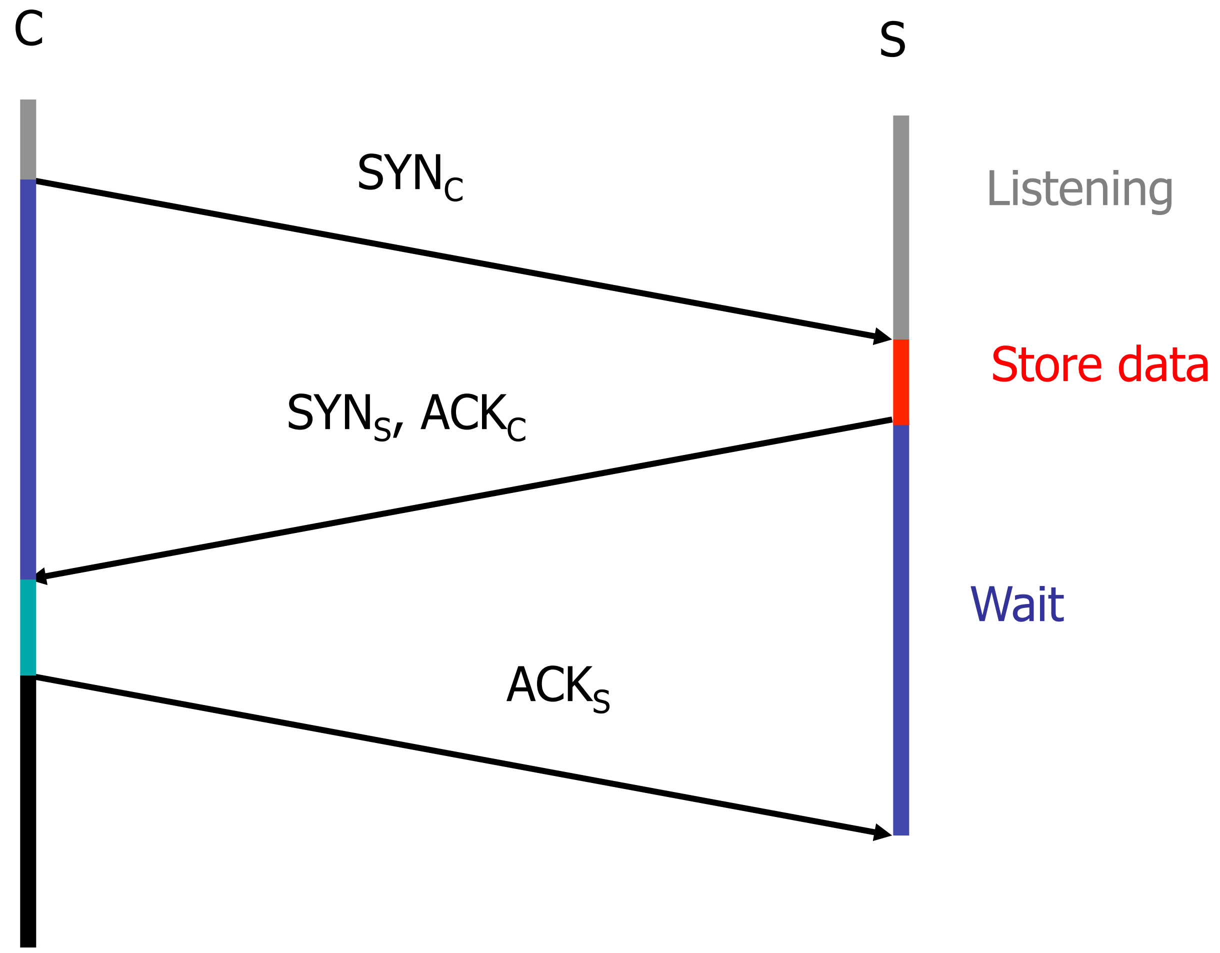
Downlink bandwidth

Memory
(e.g. TCP TCB exhaustion)

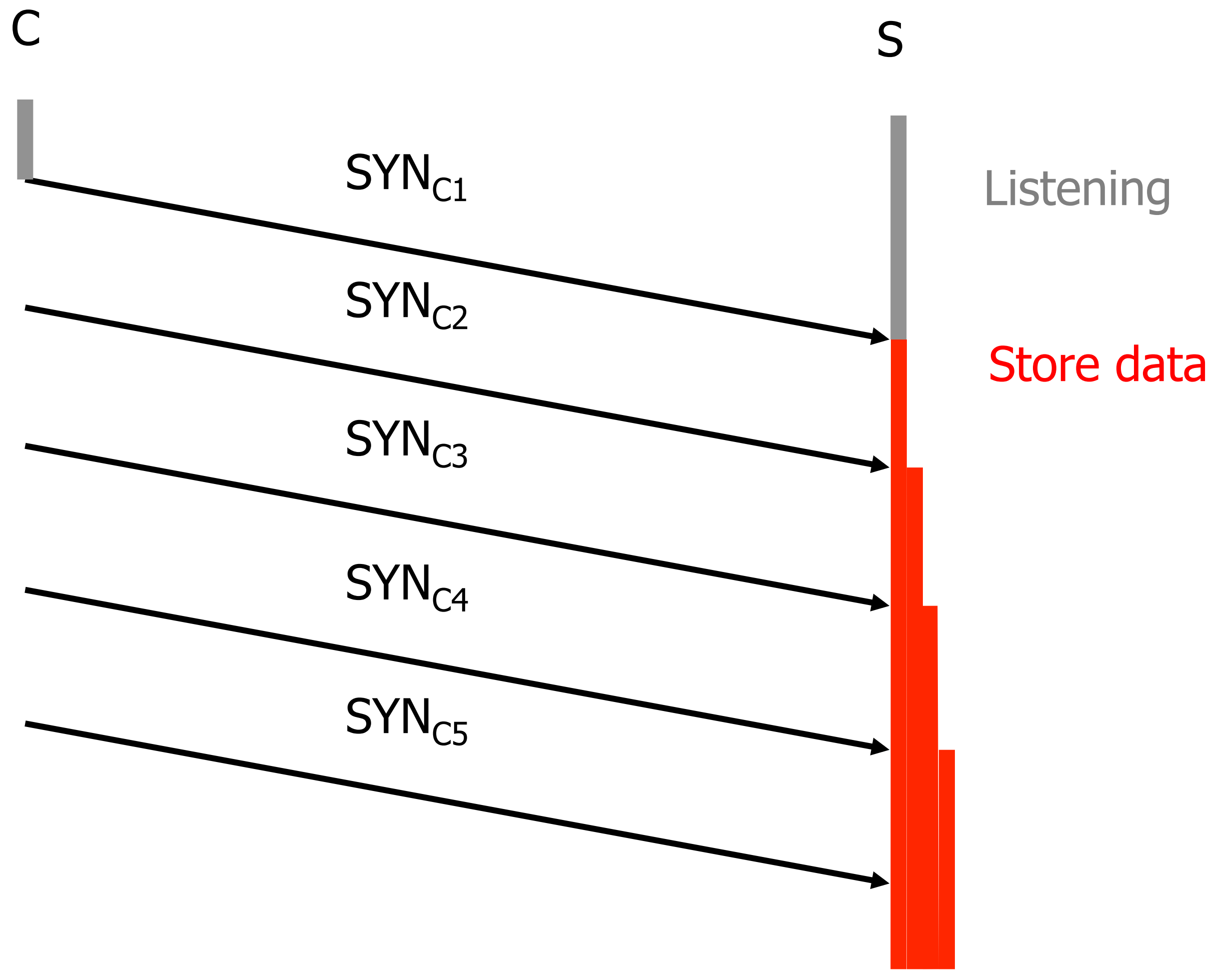
DoS: Via Resource Exhaustion

- Uplink bandwidth
  - Saturate uplink bandwidth using legitimate requests (e.g. download large image)
  - Solution: use a CDN (Akamai)
  - Solution: admission control at the server (not a network problem ??)
- CPU time similar to above
- Victim memory
  - TCP connections require state, can try to exhaust
  - E.g. SYN Flood (next few slides) (maybe a networking problem ...)

TCP Handshake



Example: SYN Flooding



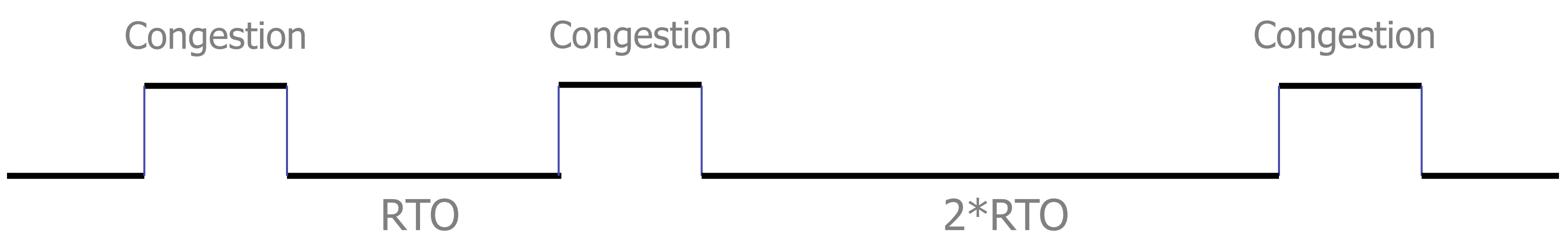
Protection against SYN Attacks

- SYN Cookies [Bernstein, Schenk]
  - Client sends SYN
  - Server responds to Client with SYN-ACK cookie
    - sqn = f(src addr, src port, dest addr, dest port, rand)
    - Server does not save state
  - Honest client responds with ACK(sqn)
  - Server checks response
    - If matches SYN-ACK, establishes connection
- SYN caching [Lemmon]
  - Doesn’t work very well ...

Other “Networking” DoS Attacks

- Attacker guesses TCP seq. number for an existing connection:
  - Attacker can send Reset packet to close connection. Results in DoS.
  - Most systems allow for a large window of acceptable seq. #’ s
  - Only have to a land a packet in
  - Attack is most effective against long lived connections, e.g. BGP.

- Congestion control DoS attack



  - Generate TCP flow to force target to repeatedly enter retransmission timeout state
  - Difficult to detect because packet rate is low

On to the Paper