0x1A Great Papers in Computer Security

Vitaly Shmatikov

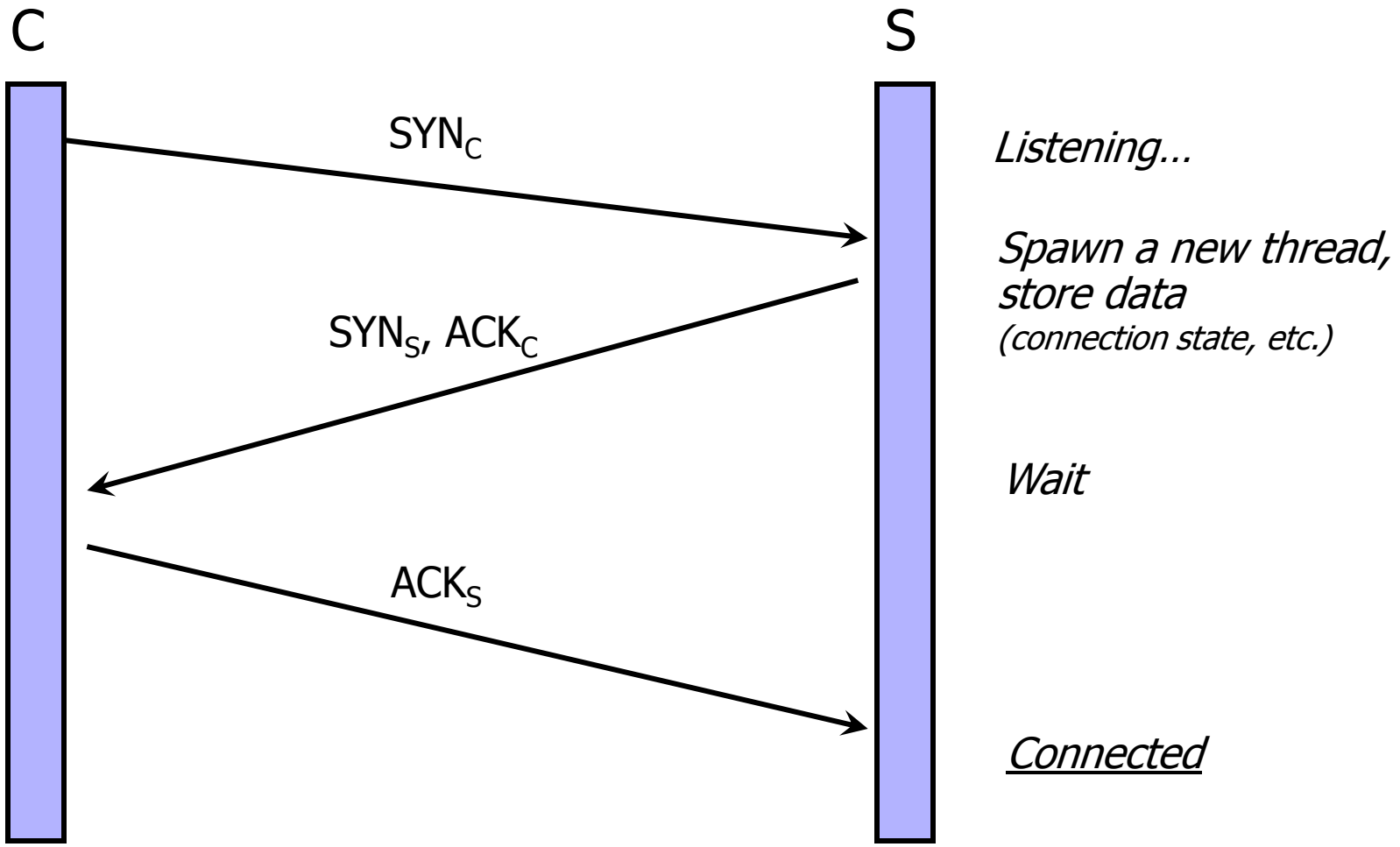
http://www.cs.utexas.edu/~shmat/courses/cs380s/

D. Bernstein

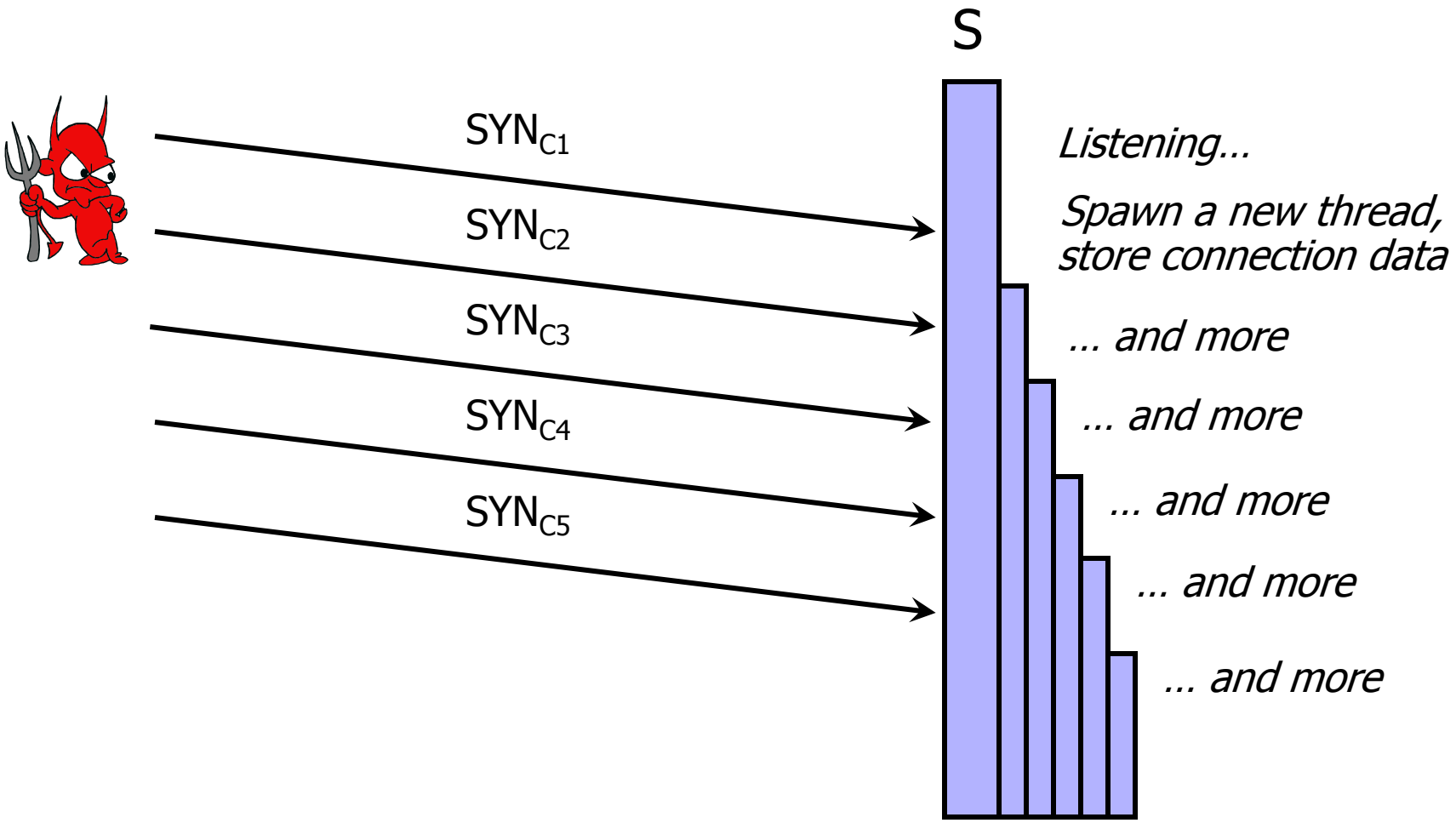
SYN cookies

(1996)

TCP Handshake



SYN Flooding Attack



SYN Flooding Explained

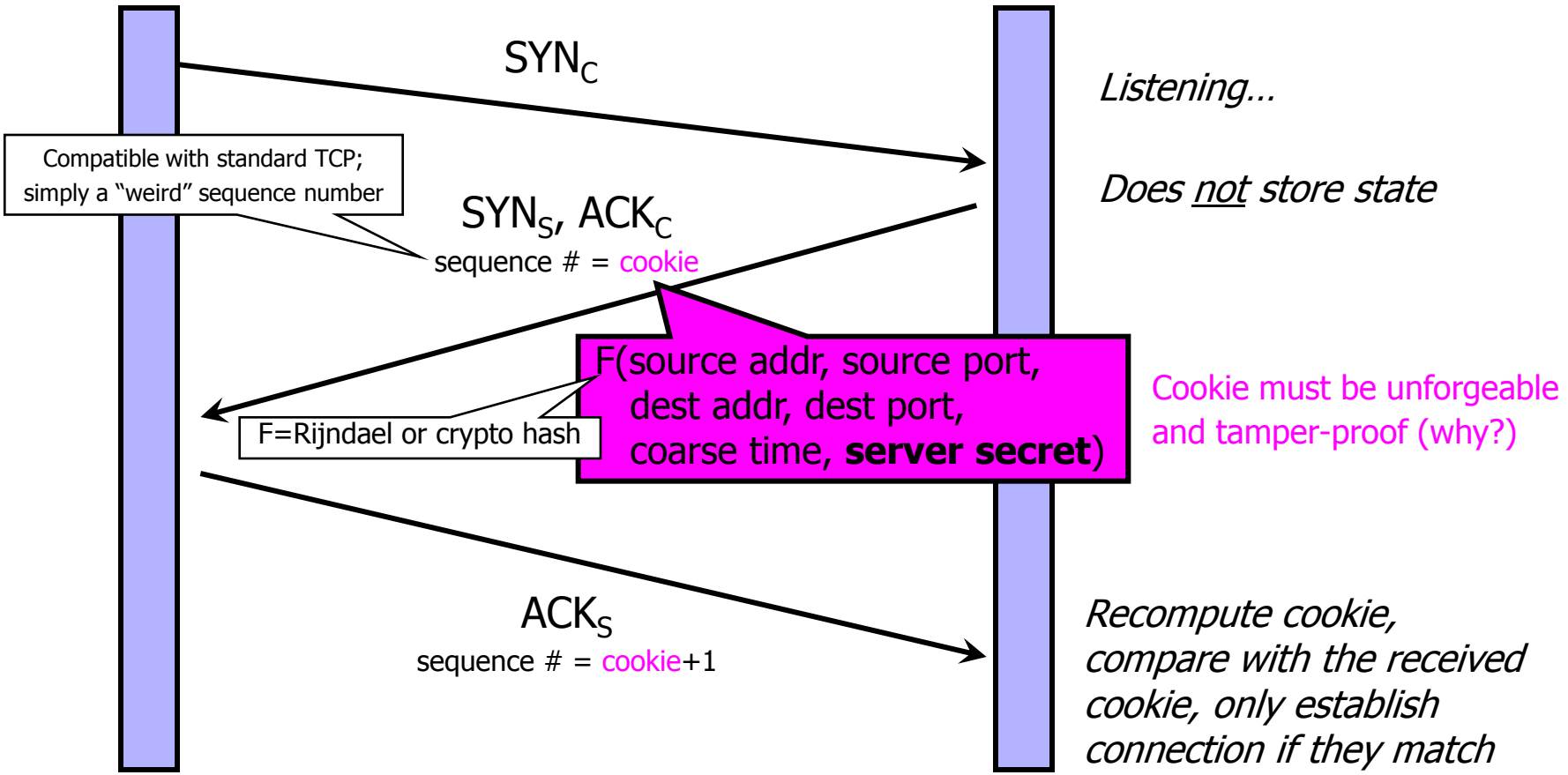
- Attacker sends many connection requests with spoofed source addresses
- Victim allocates resources for each request
  - New thread, connection state maintained until timeout
  - Fixed bound on half-open connections
- Once resources exhausted, requests from legitimate clients are denied
- This is a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

Preventing Denial of Service

- DoS is caused by asymmetric state allocation
  - If responder opens new 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 two 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 and Schenk]



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 responds with hashed connection data instead of message #2
  - Client confirms by returning hashed data
    - If source IP address is spoofed, attacker can't confirm
  - Need an extra step to send postponed message #2, except in TCP (SYN-ACK already there)