CS 5435

Attacks on TCP/IP
Denial of Service

Vitaly Shmatikov

Warm Up: 802.11b

NAV (Network Allocation Vector)

- 15-bit field, max value: 32767
- Any node can reserve channel for NAV microseconds
- No one else should transmit during NAV period

  … but not followed by most 802.11b cards

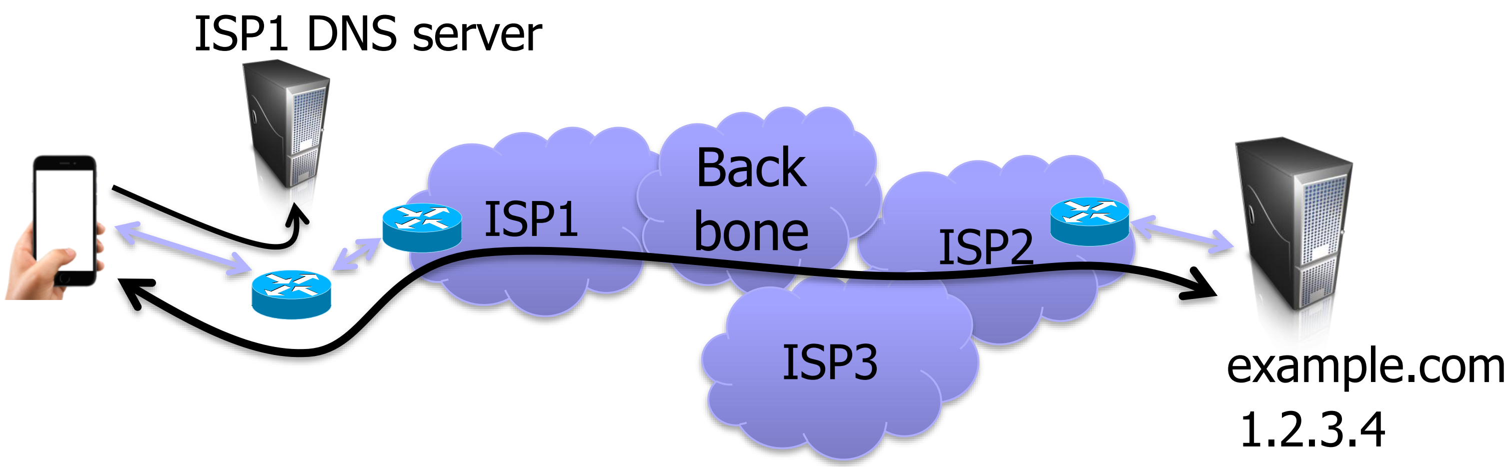
De-authentication

- Any node can send deauth packet to AP
- Deauth packet unauthenticated

  … attacker can repeatedly deauth anyone

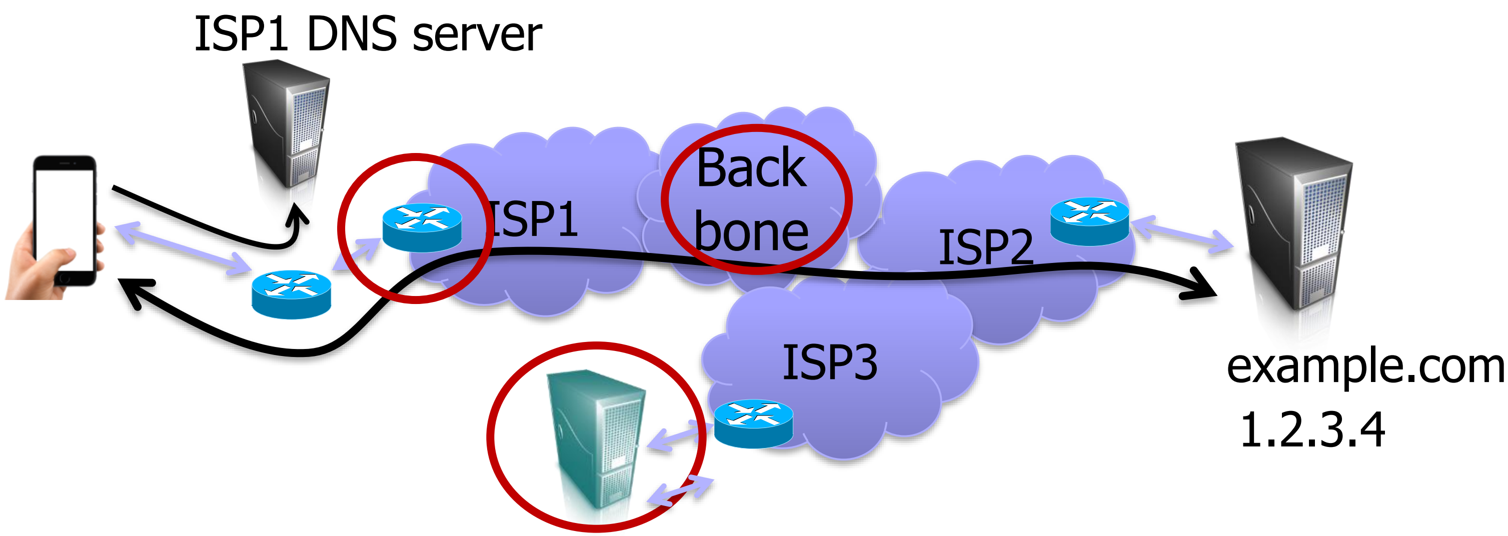
Steps to Send an HTTP Request

Pre HTTP/3



1. DNS lookup on example.com to get IP address (1.2.3.4)
2. TCP connection setup via 4-way handshake of IP packets to and from 1.2.3.4
3. Send HTTP request over TCP connection

Network Threat Models



(1) Malicious hosts } Off-path attackers

On-path attackers { (2) Subverted routers or links
(3) Malicious ISPs or backbone

Network Attacks

BGP and IP hijacking

DNS cache poisoning

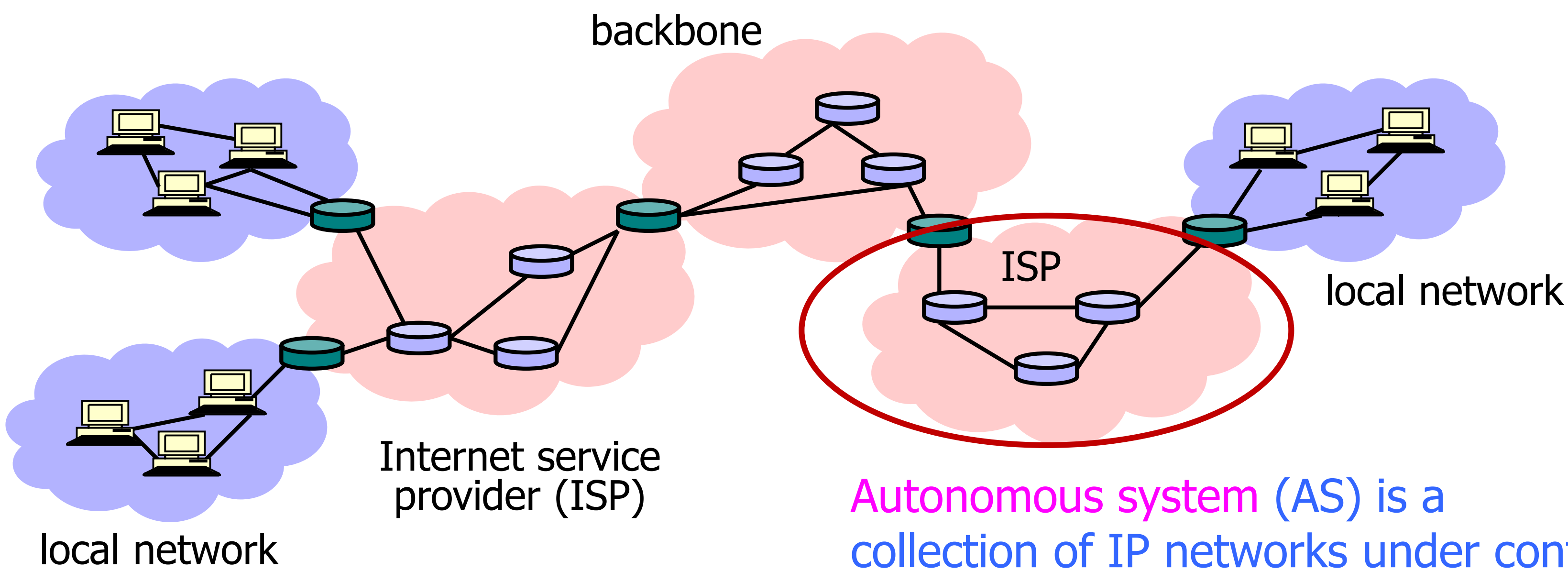
IP spoofing

- Simple untraceable DoS attacks

Off-path TCP injection

- Allows injecting traffic into other connections

Internet Is a Network of Networks



Autonomous system (AS) is a collection of IP networks under control of a single administrator (e.g., ISP)

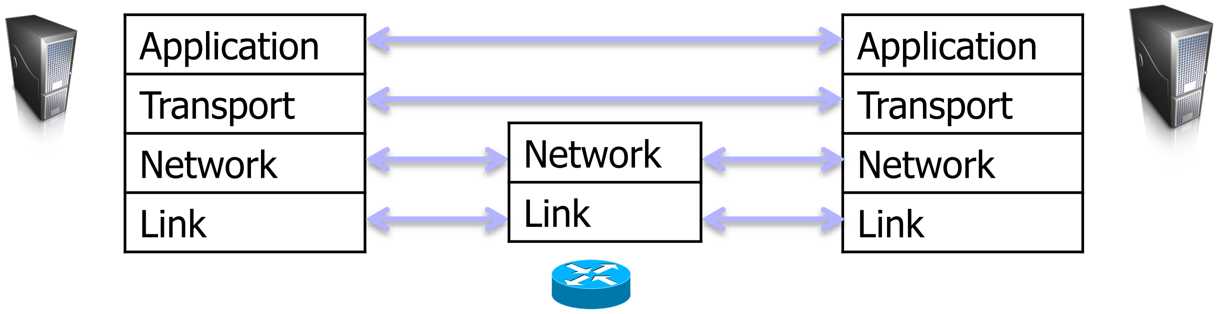
TCP/IP for packet routing and connections

Border Gateway Protocol (BGP) for route discovery

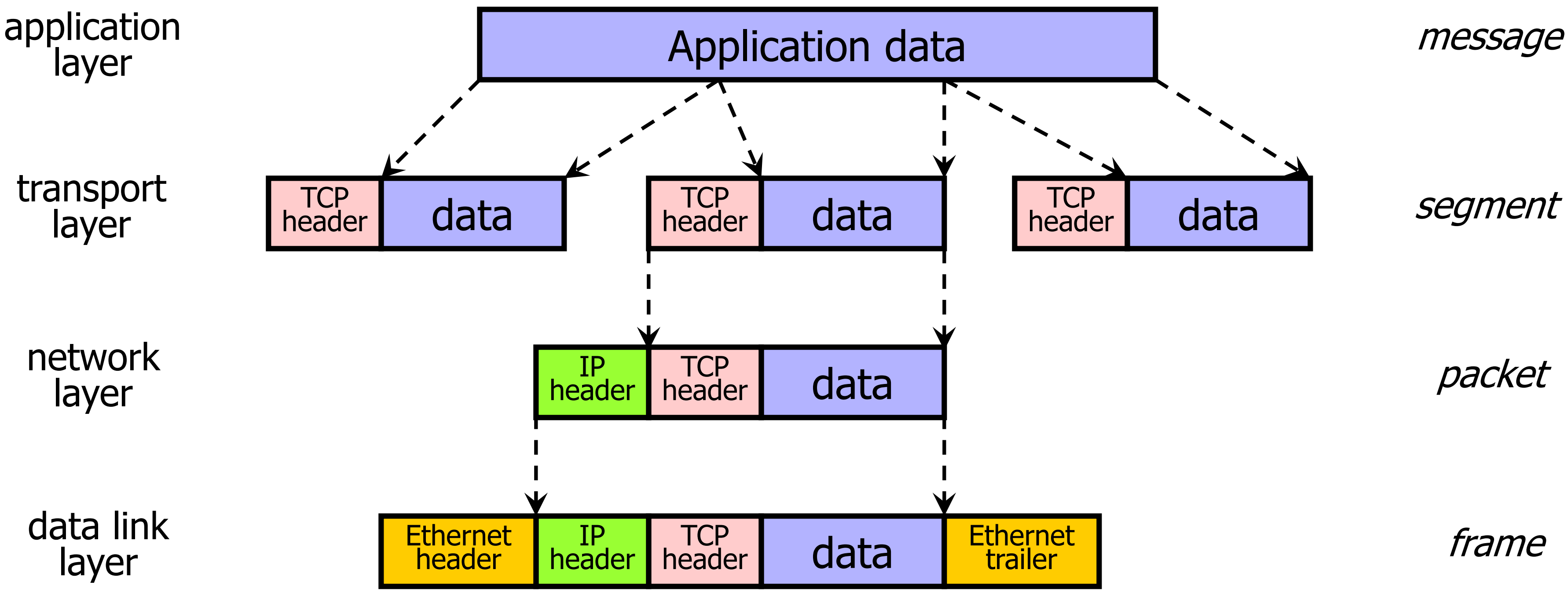
Domain Name System (DNS) for IP address discovery

Internet Protocol Stack

Application	HTTP, DNS, FTP, SMTP, SSH, etc.
Transport	TCP, UDP
Network	IP, ICMP, …
Link	802x (802.11, Ethernet)



Data Formats



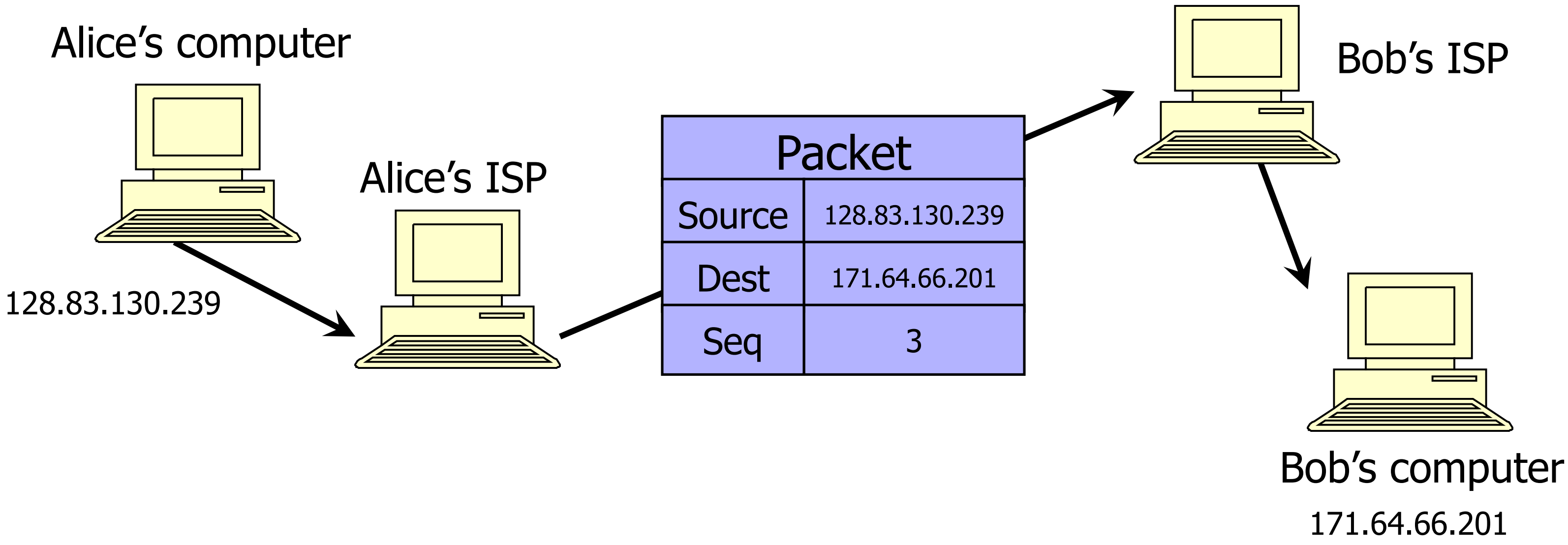
IP (Internet Protocol)

Connectionless

- Unreliable, "best-effort" protocol

Uses numeric addresses for routing

Typically several hops in the route



TCP (Transmission Control Protocol)

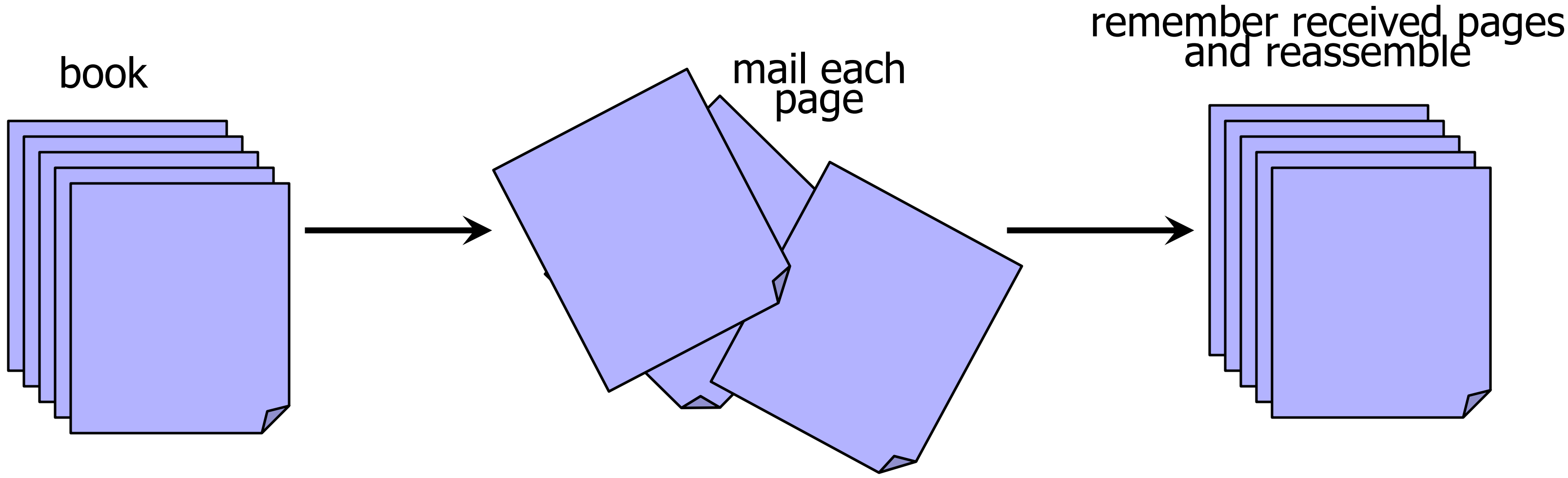
Sender: break data into packets

- 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



ICMP (Control Message Protocol)

Provides feedback about network operation

- "Out-of-band" messages carried in IP packets

Error reporting, congestion control, reachability…

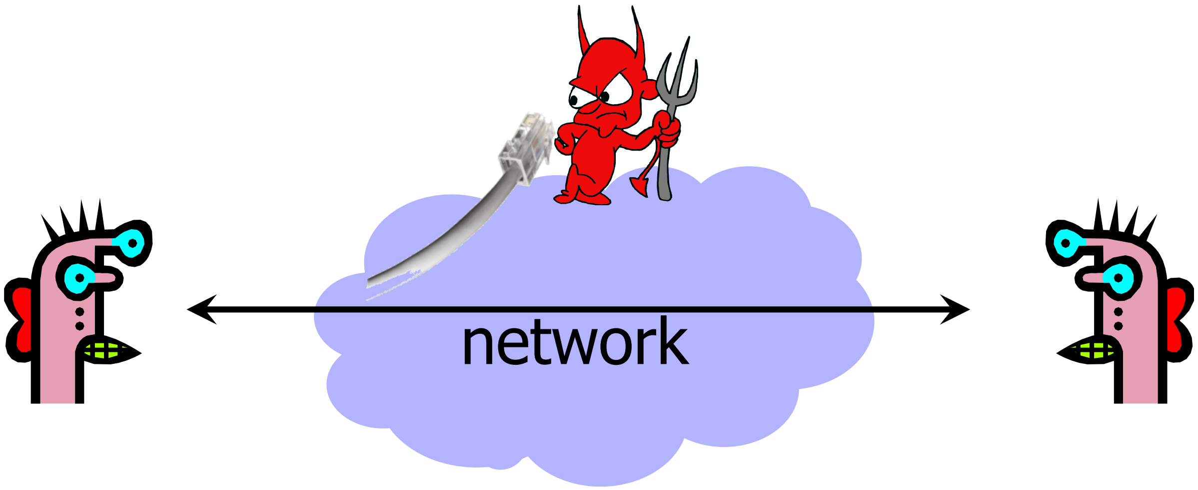
- Destination unreachable
- Time exceeded
- Parameter problem
- Redirect to better gateway
- Reachability test (echo / echo reply)
- Message transit delay (timestamp request / reply)

Packet Sniffing

Many applications send data unencrypted

- ftp, telnet send passwords in the clear

Network interface card (NIC) in "promiscuous mode" reads all passing data



Solution: encryption (e.g., IPsec, HTTPS), improved routing

“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

“Teardrop” and “Bonk”

TCP fragments contain Offset field

Attacker sets Offset field to overlapping values

- Bad implementation of TCP/IP will crash when attempting to re-assemble the fragments

… or to very large values

- Bad TCP/IP implementation will crash

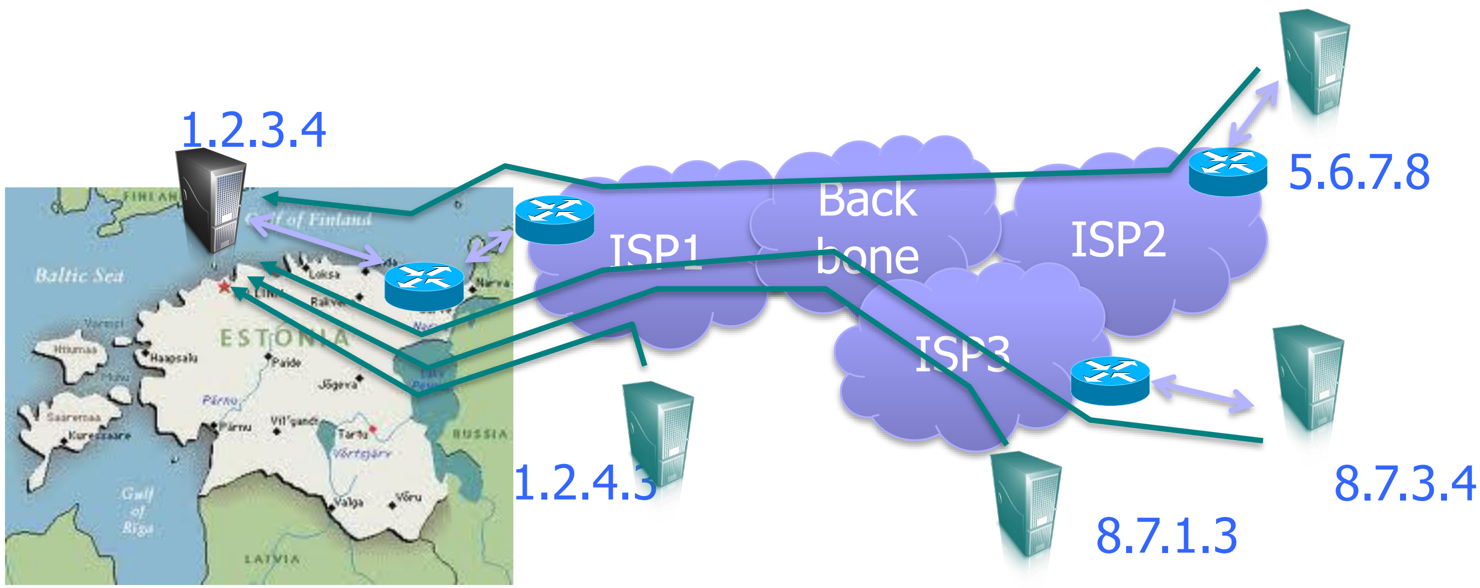
Solution: use up-to-date TCP/IP implementation

"LAND"

- IP packet with source address, port equal to destination address, port; SYN flag set
- Triggers loopback in the Windows XP SP2 implementation of TCP/IP stack, locks up CPU

Solution: ingress filtering

DDoS Attack on Estonia



April 27, 2007

Continued for weeks, with varying levels of intensity

Government, banking, news, university websites

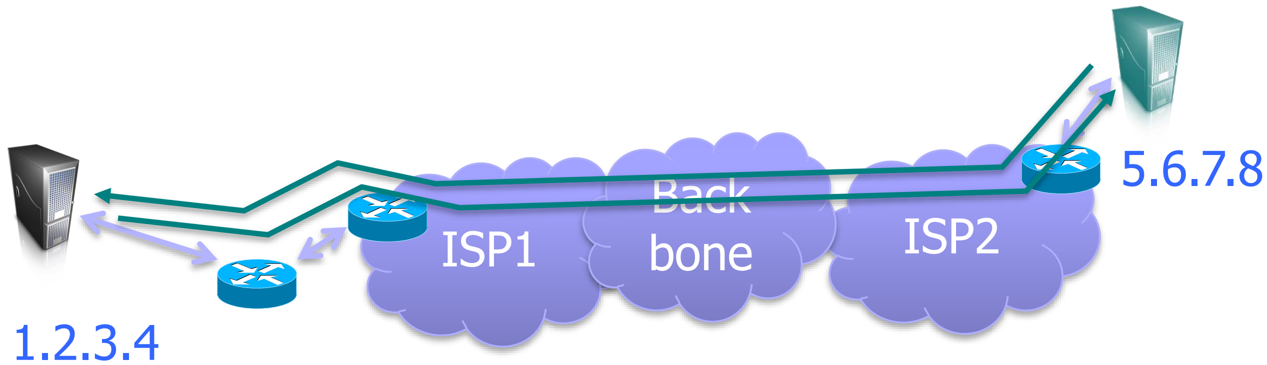
Government shut down international Internet connections

Telegram blames China for ‘powerful DDoS attack’ during Hong Kong protests

Telegram CEO says ‘IP addresses coming mostly from China’ were to blame

By Jon Porter | @JonPorty | Jun 13, 2019, 4:21am EDT

DoS Attack?

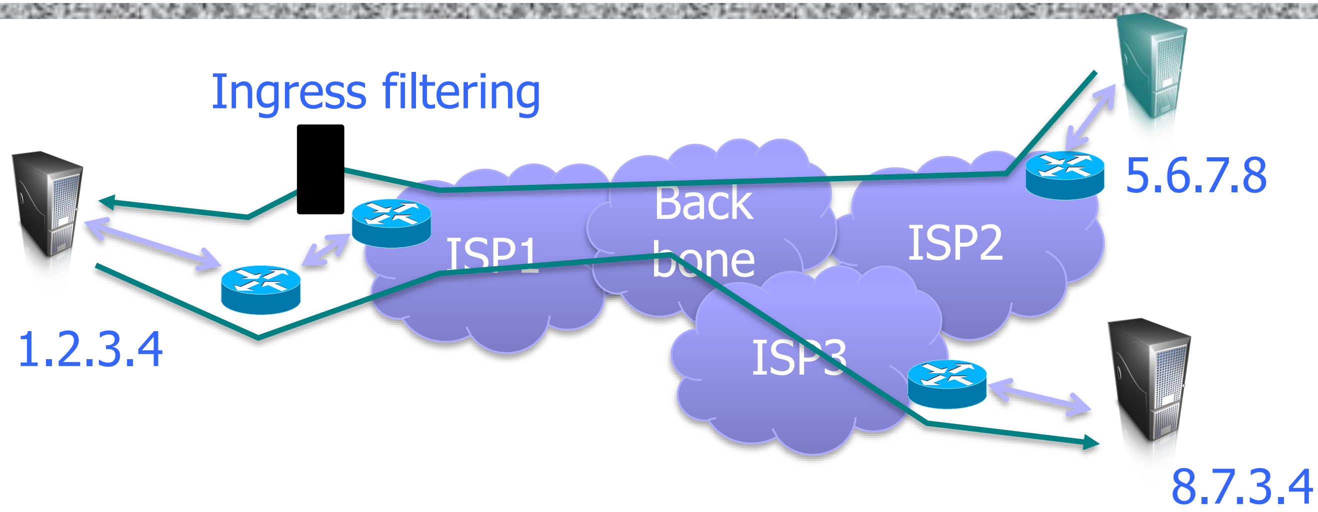


Goal: prevent legitimate users from accessing victim (1.2.3.4)

ICMP ping flood

- Attacker sends ICMP pings as fast as possible to victim
- When will this work as a DoS? Attacker resources > victim's
- How can this be prevented? Ingress filtering of attacker IP addresses near victim once attack identified

Avoiding Ingress Filtering



1. Attacker can send packet with fake ("spoofed") source IP address. Packet will get routed correctly. Replies will not

   Send IP packet with source: 8.7.3.4 dest: 1.2.3.4 from 5.6.7.8

2. Distribute attack across many IP addresses

Amplification

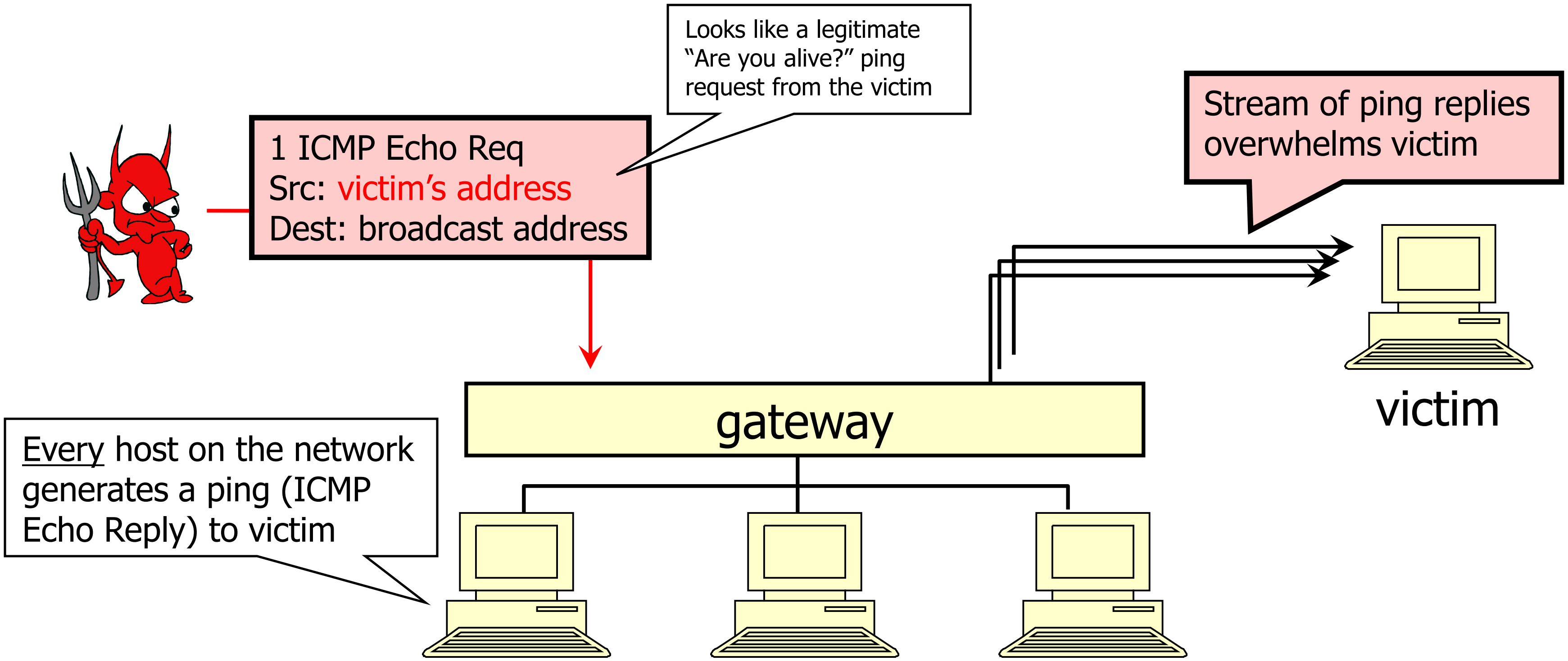
Key element of powerful DoS attack

Achieves attacker resources >>> victim's

- 1 request sent by attacker => N requests received by the victim
- 1 byte sent by attacker => N bytes received by the victim
  - N can be 200+ in some attacks!

- Computational time
- Logical resources on target server

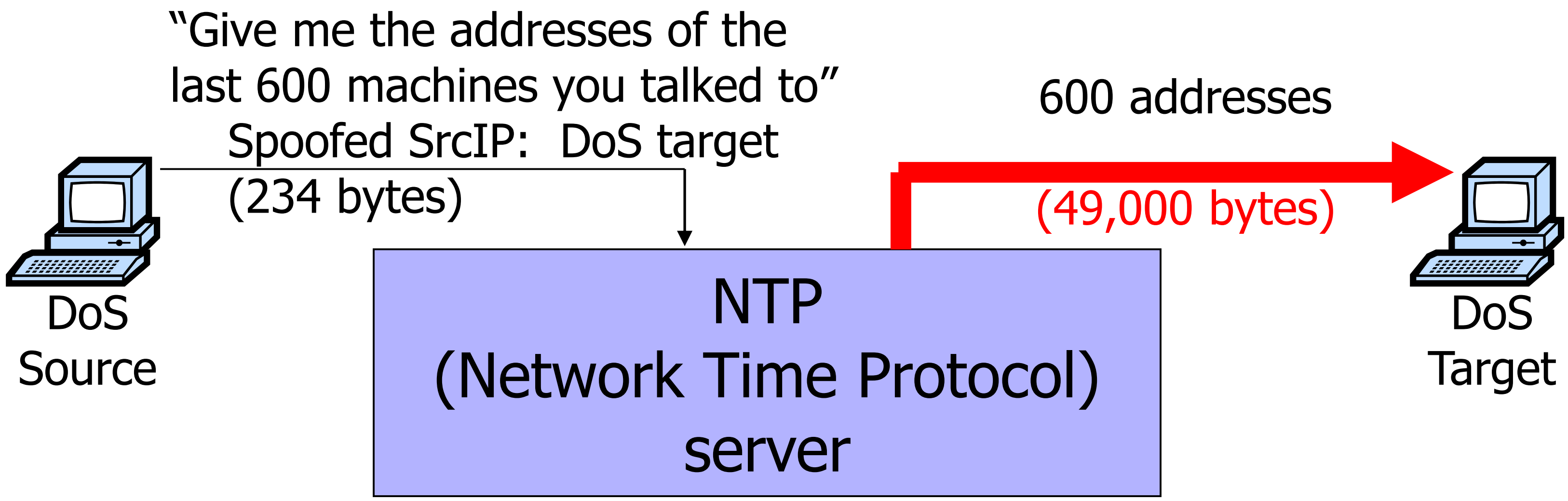
“Smurf” Reflector Attack



Solution: reject external packets to broadcast addresses

NTP Amplification Attack

x206 amplification



December 2013 – February 2014:
400 Gbps DDoS attacks involving 4,529 NTP servers

7 million unsecured NTP servers on the Internet (Arbor)

Memcached DDoS Attacks

Memcached is a popular in-memory data store

Supports UDP requests

Standard reflector attack

- Insert data into the Memcached server
- Send UDP requests with source IP addr of victim

Relies on vulnerable memcached servers

- Default configuration: accept UDP requests from anywhere

DDoS Attack on GitHub

https://github.blog/2018-03-01-ddos-incident-report/

Feb 2018 attack against GitHub

- **51,000x** bandwidth magnification
- 1.3 TB/s of traffic to GitHub from 1000+ ASes
- GitHub offline for 5 minutes

Response?

- Use BGP announcement (what's that?) to route GitHub traffic through Akamai
- Akamai gives more capacity + helps filter out bogus requests
- Turn off UDP support in Memcached (now off by default)

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

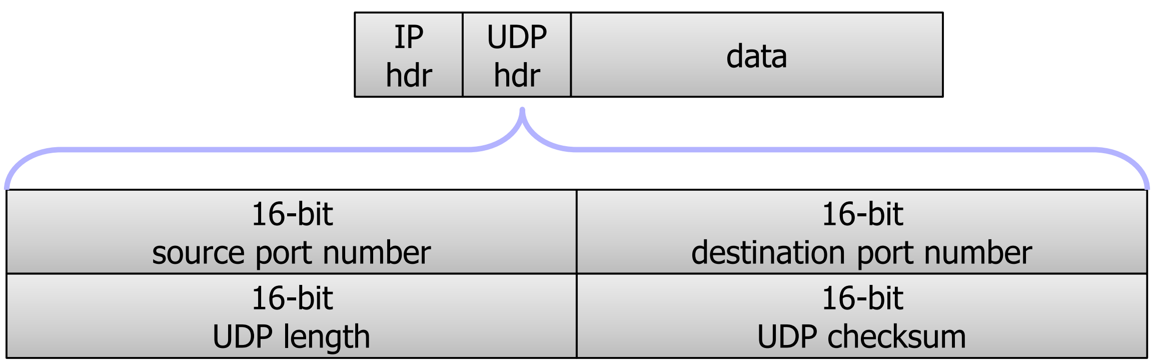
No acknowledgement, no flow control, no message continuation

Denial of service by UDP data flood

Why UDP in Reflection Attacks?

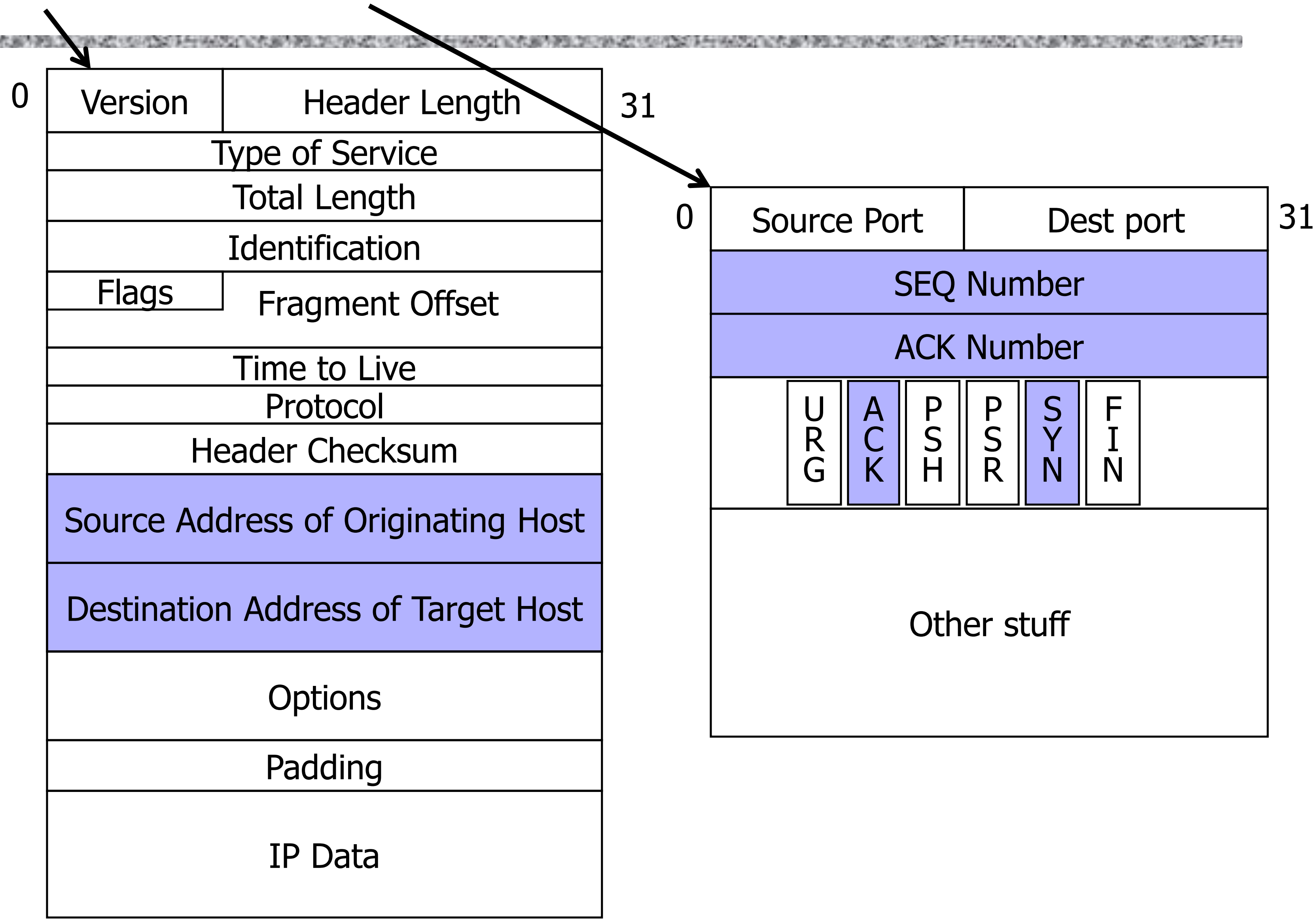
DNS, memcached, … application-layer protocols running on UDP often exploited in DoS attacks

Single packet to victim service yields response, so spoofing works

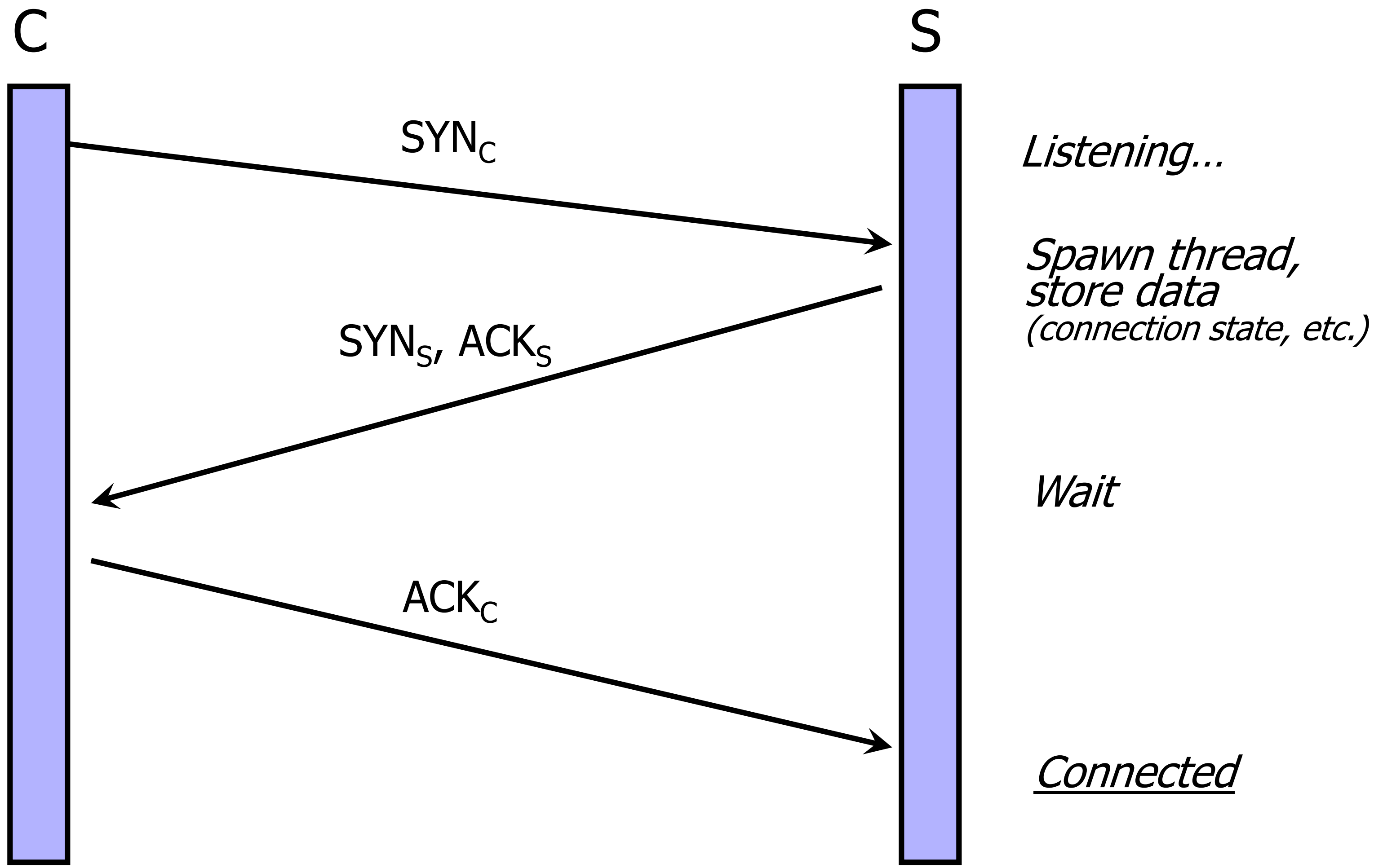


length = header len + data len

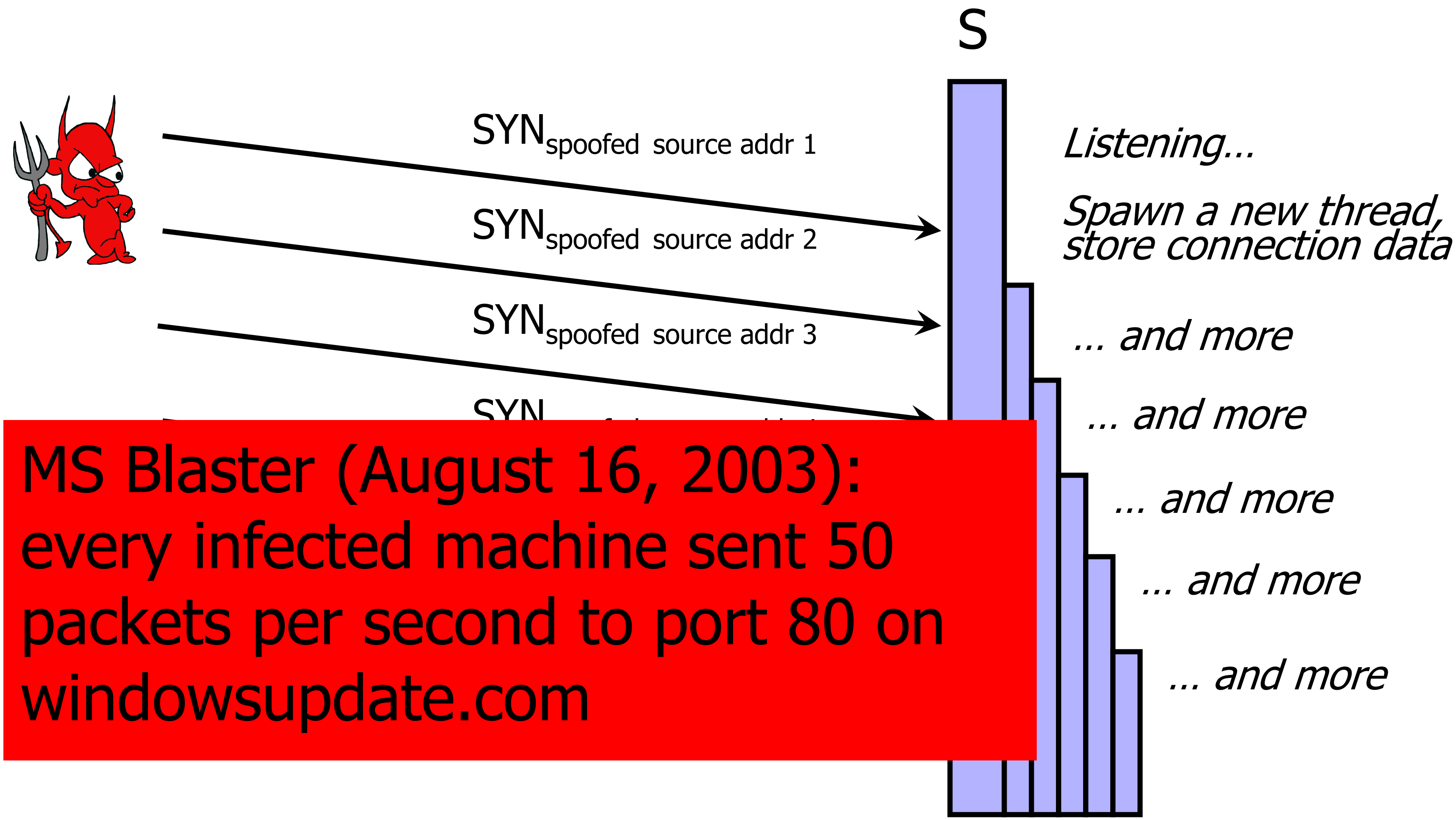
IP and TCP Headers



TCP Handshake



SYN Flooding Attack



MS Blaster (August 16, 2003): every infected machine sent 50 packets per second to port 80 on windowsupdate.com

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 pattern

- It costs nothing to TCP initiator to send a connection request, but TCP responder must spawn a thread for each request - asymmetry!

Low-Rate SYN Floods

[Phrack 48, no 13, 1996]

OS	Backlog queue size
Linux 1.2.x	10
FreeBSD 2.1.5	128
WinNT 4.0	6

Backlog timeout: 3 minutes

Attacker need only send
128 SYN packets every 3 minutes

$\Rightarrow$ low-rate SYN flood

Preventing SYN Floods

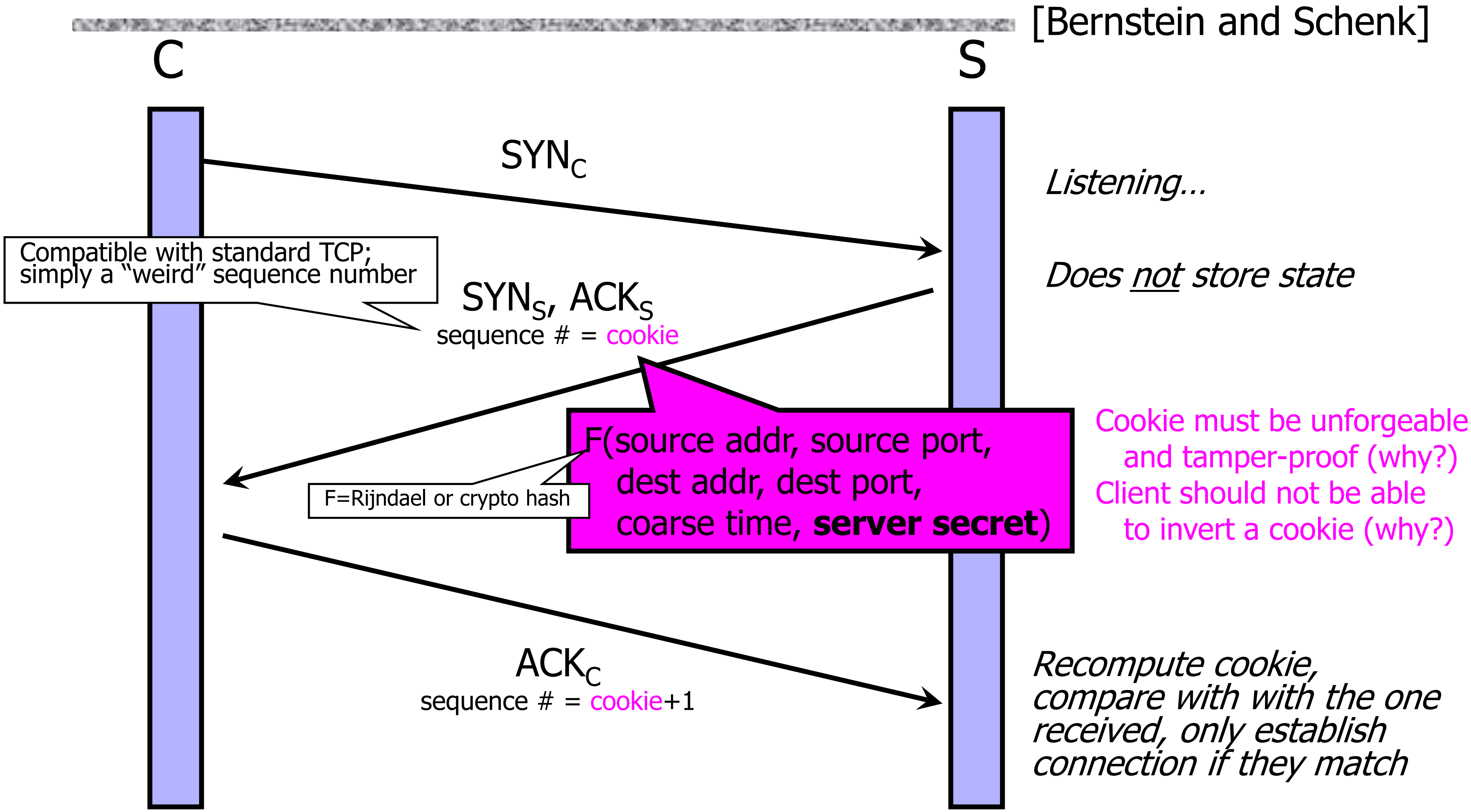
DoS is caused by asymmetric state allocation

- 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



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

Anti-Spoofing Cookies: Basic Pattern

Client sends request (message #1) to server

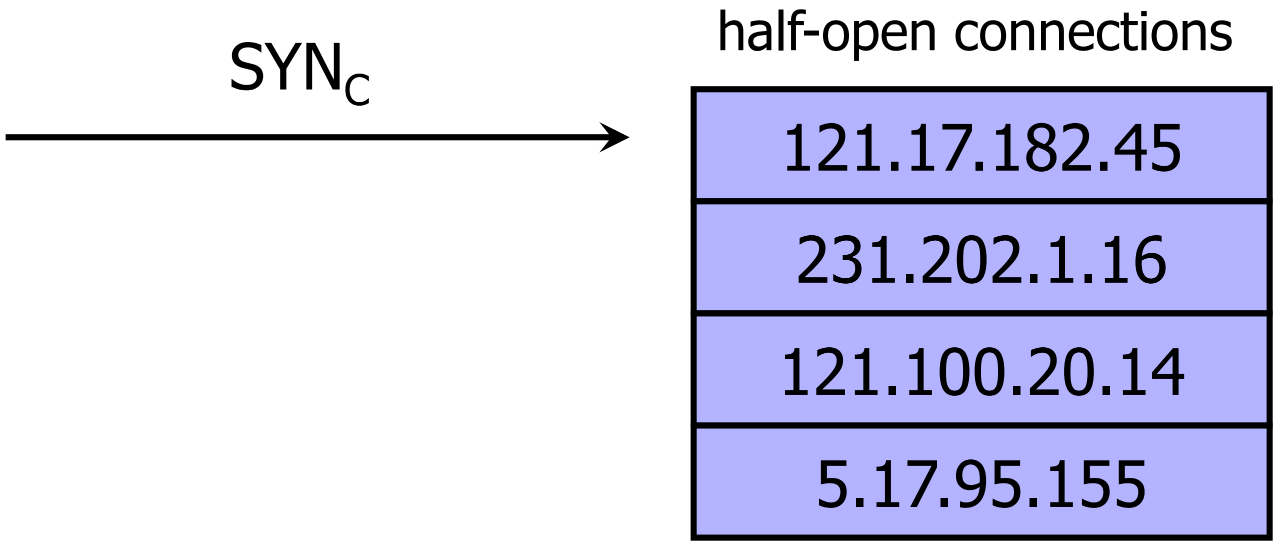
Typical protocol:

- 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 bogus, attacker can't confirm
- Need an extra step to send postponed message #2, <u>except</u> in TCP (can piggyback on SYN-ACK in TCP)

Another Defense: Random Deletion



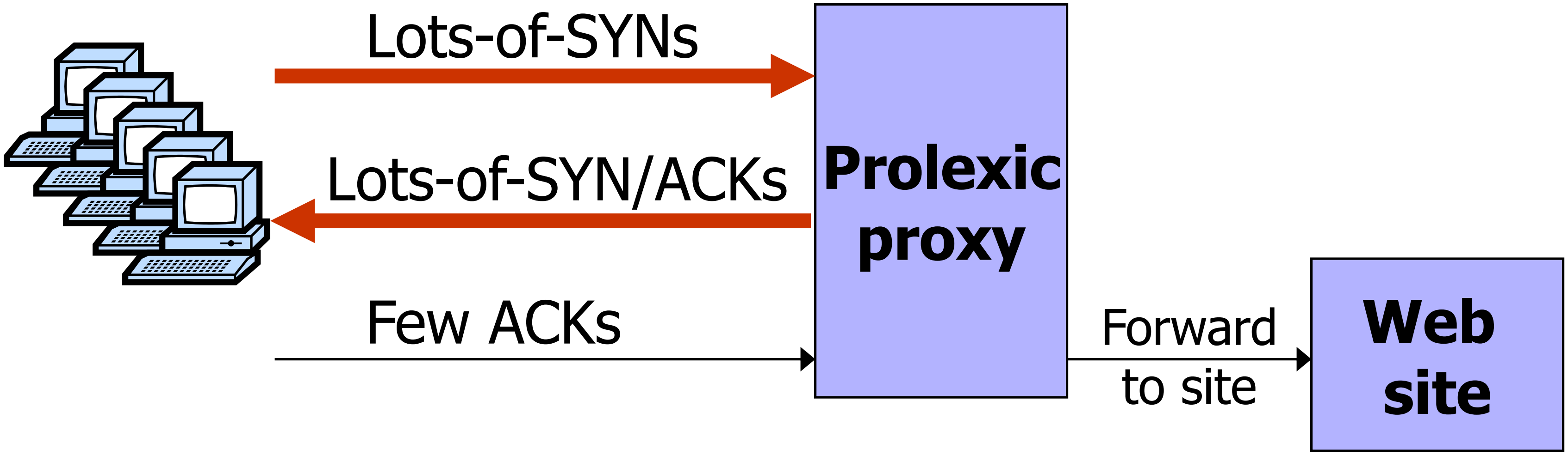
If SYN queue is full, delete random entry

- Legitimate connections have a chance to complete
- Fake addresses will be eventually deleted

Easy to implement

Prolexic

Idea: only forward established TCP connections to site



Prolexic purchased by Akamai in 2014
Many companies: Cloudflare, Imperva, Arbor Networks, …

Other Junk-Packet Attacks

Attack Packet	Victim Response	Rate: attk/day [ATLAS 2013]
TCP SYN to open port	TCP SYN/ACK	773
TCP SYN to closed port	TCP RST	
TCP ACK or TCP DATA	TCP RST	
TCP RST	No response	
TCP NULL	TCP RST	
ICMP ECHO Request	ICMP ECHO Response	**50**
UDP to closed port	ICMP Port unreachable	387

Proxy must keep floods of these away from website

Stronger Attack: TCP Con Flood

Command bot army to:

- Complete TCP connection to web site
- Send short HTTP HEAD request
- Repeat

Will bypass SYN flood protection proxy but …

- Attacker can no longer use random source IPs
  - Reveals location of bot zombies
- Proxy can now block or rate-limit bots

TCP Connection Spoofing

Each TCP connection has associated state

- Sequence number, port number

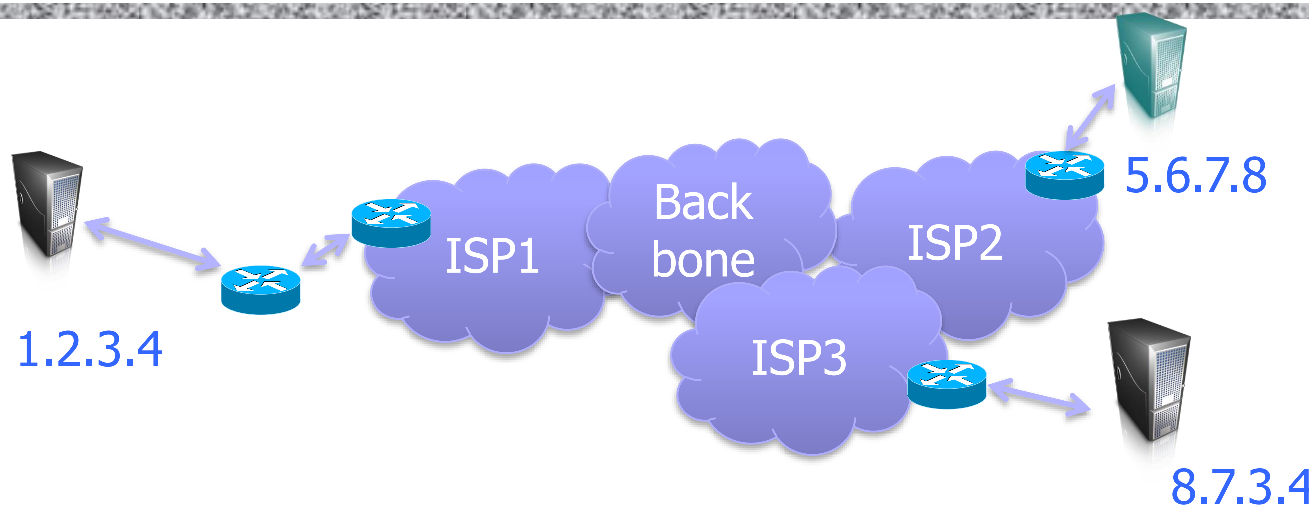
TCP state is easy to guess

- Port numbers standard, seq numbers predictable

Can inject packets into existing connections

- If attacker knows initial sequence number and amount of traffic, can guess likely current number
- Guessing a 32-bit seq number is not practical, BUT…
- Most systems accept large windows of sequence numbers (to handle packet losses), so send a flood of packets with likely sequence numbers

Predictable Sequence Numbers



4.4 BSD used predictable initial sequence numbers (ISNs)

- At system initialization, set ISN to 1
- Increment ISN by 64,000 every half-second

What can a clever attacker do?
(assume spoofing possible)

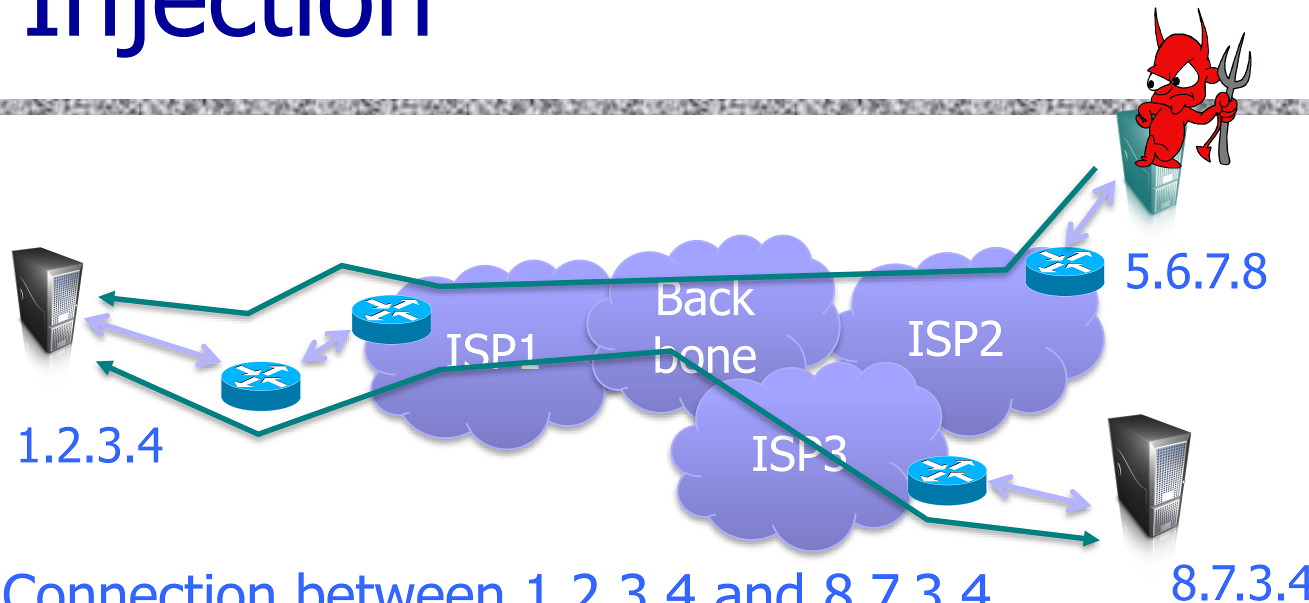
DoS by Connection Reset

If attacker can guess the current sequence number for an existing connection, can send Reset packet to close it

Especially effective against long-lived connections

- For example, BGP route updates

TCP Injection



Connection between 1.2.3.4 and 8.7.3.4

Forge a FIN packet from 8.7.3.4 to 1.2.3.4

src: 8.7.3.4
dst: 1.2.3.4

seq#(8.7.3.4)
FIN

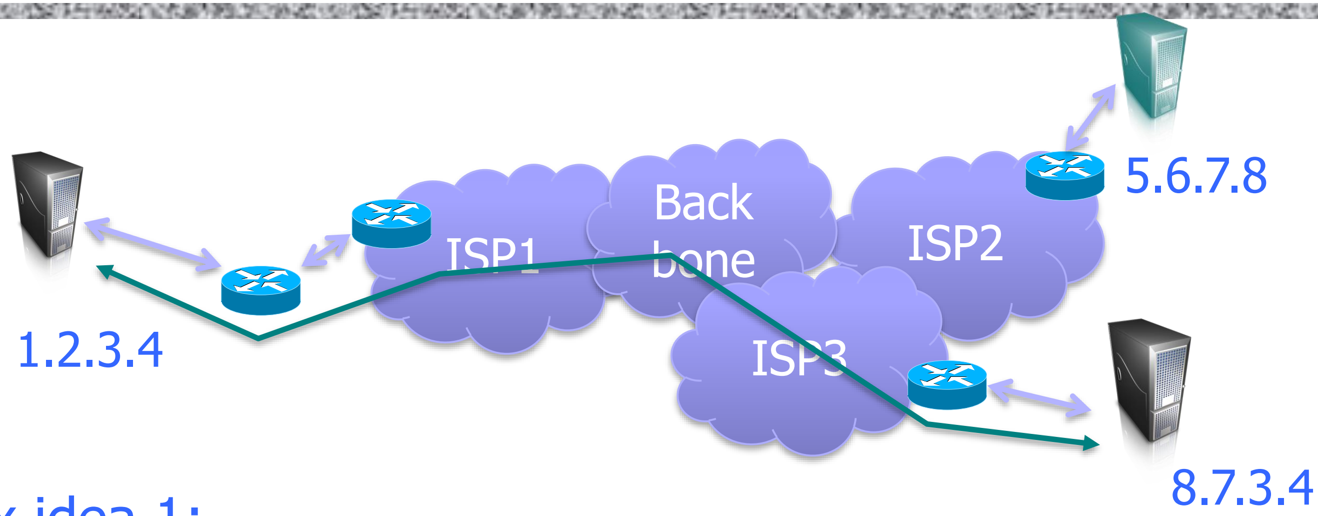
Forge some application-layer packet from 8.7.3.4 to 1.2.3.4

src: 8.7.3.4
dst: 1.2.3.4

seq#(8.7.3.4)
"rsh rm –rf /"

Attacker can't see server's responses, but can bypass IP address-based authentication (remote shell, SPF defense against spam)

Fixing Predictable Seq Numbers



Fix idea 1:

- Random ISN at system startup
- Increment by 64,000 each half second

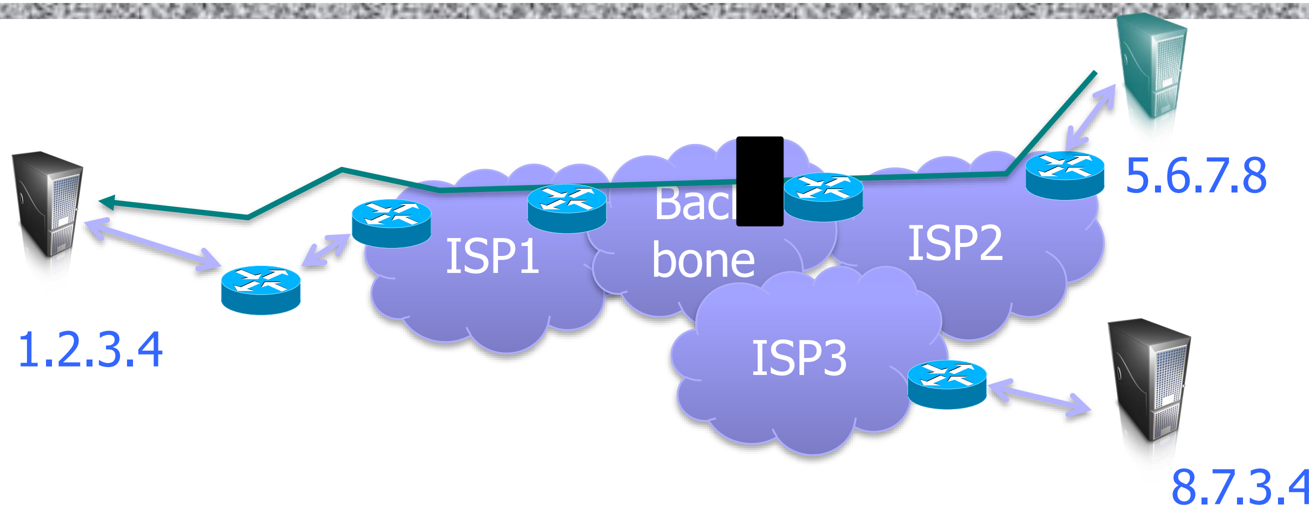
Better fix:

- Random ISN for every connection

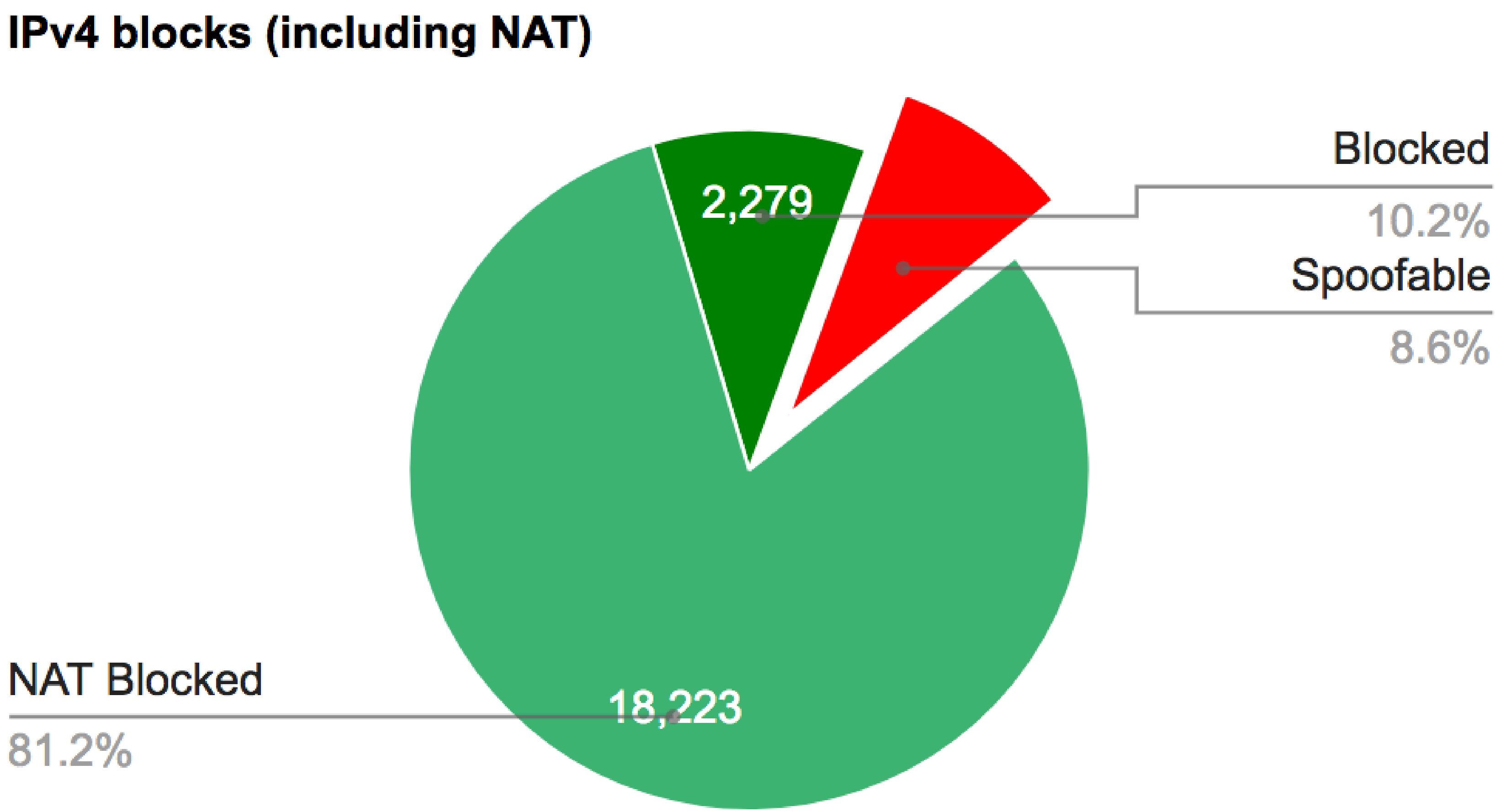
Remains an issue in some cases:

- Any FIN accepted with seq# in receive window: 2^{17} attempts
- Side-channel attacks to infer seq#

Fixing Spoofing



- IP traceback: techniques for inferring actual source of a (spoofed) packet
- BCP 38 (RFC 2827): upstream ingress filtering to drop spoofed packets
  - Ideally, all network traffic providers would perform ingress filtering



https://spoofer.caida.org/summary.php

Other Countermeasures

Above transport layer: Kerberos

- Provides authentication, protects against application-layer spoofing
- Does <u>not</u> protect against connection hijacking

Above network layer: SSL/TLS and SSH

- Protects against connection hijacking and injected data
- Does <u>not</u> protect against DoS by spoofed packets

Network (IP) layer: IPsec

- Protects against hijacking, injection, DoS using connection resets, IP address spoofing