ECE/COMPSCI 356 Computer Network Architecture

Lecture 23: TCP Security

Neil Gong

neil.gong@duke.edu

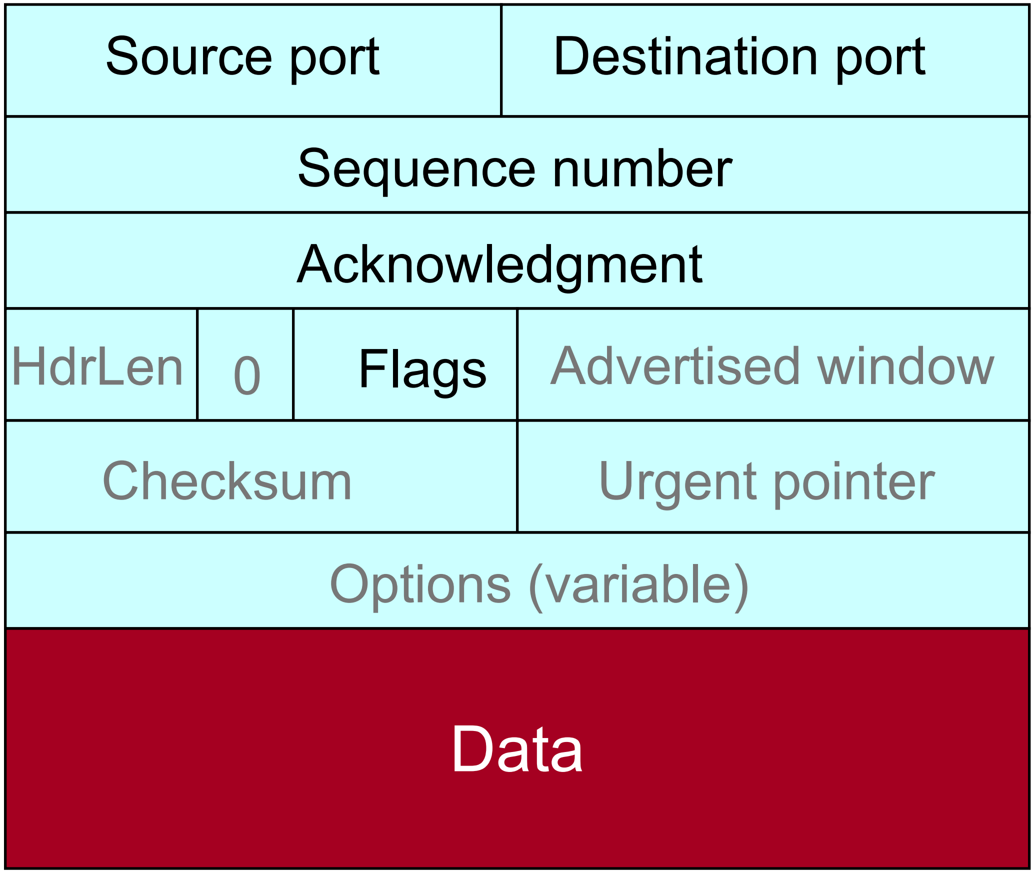
Slides credit: Vern Paxson, Dawn Song

Overview

- TCP disruption
- TCP injection
- TCP spoofing
- SYN flooding
  - DoS

TCP Threat: Disruption

- Normally, TCP finishes (“closes”) a connection by each side sending a `FIN` control message
  - Reliably delivered, since other side must ack
- But: if a TCP endpoint finds unable to continue (process dies; info from other “peer” is inconsistent), it abruptly terminates by sending a **RST** control message
  - Unilateral
  - Takes effect immediately (no ack needed)
  - Only accepted by peer if has correct sequence number



Source port

Destination port

Sequence number

Acknowledgment

HdrLen

0

Flags

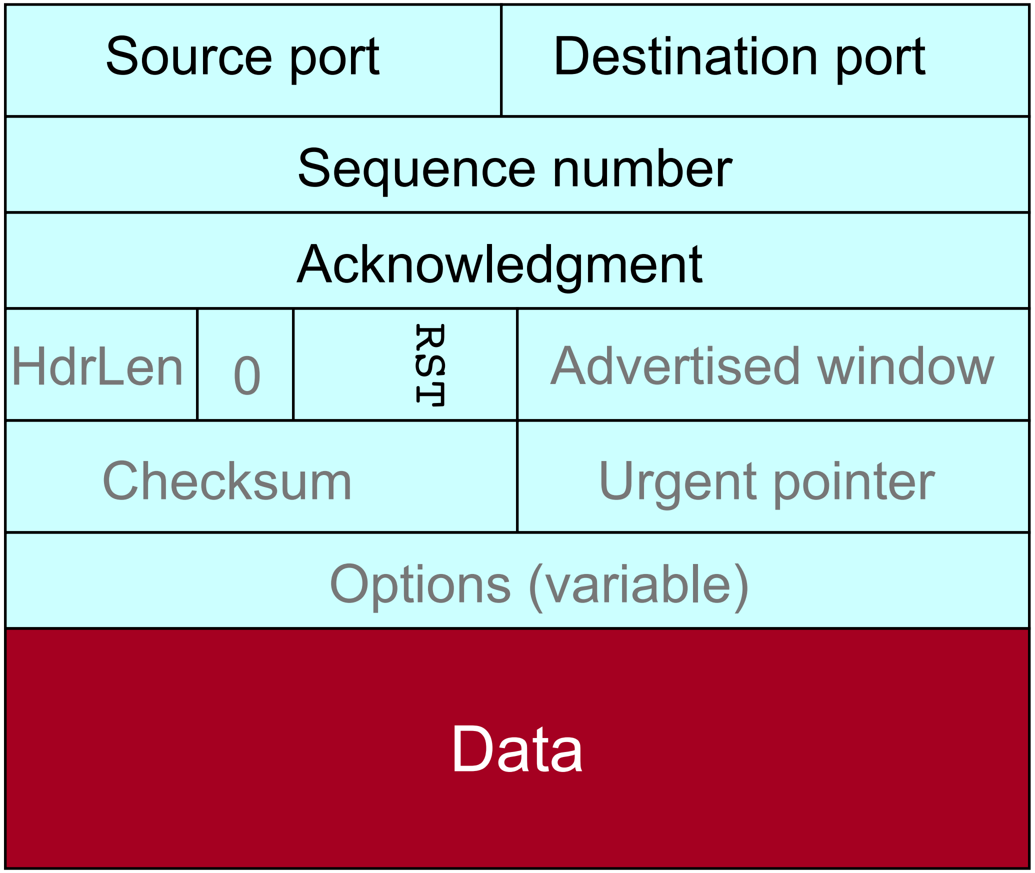
Advertised window

Checksum

Urgent pointer

Options (variable)

Data



Source port

Destination port

Sequence number

Acknowledgment

HdrLen

0

RST

Advertised window

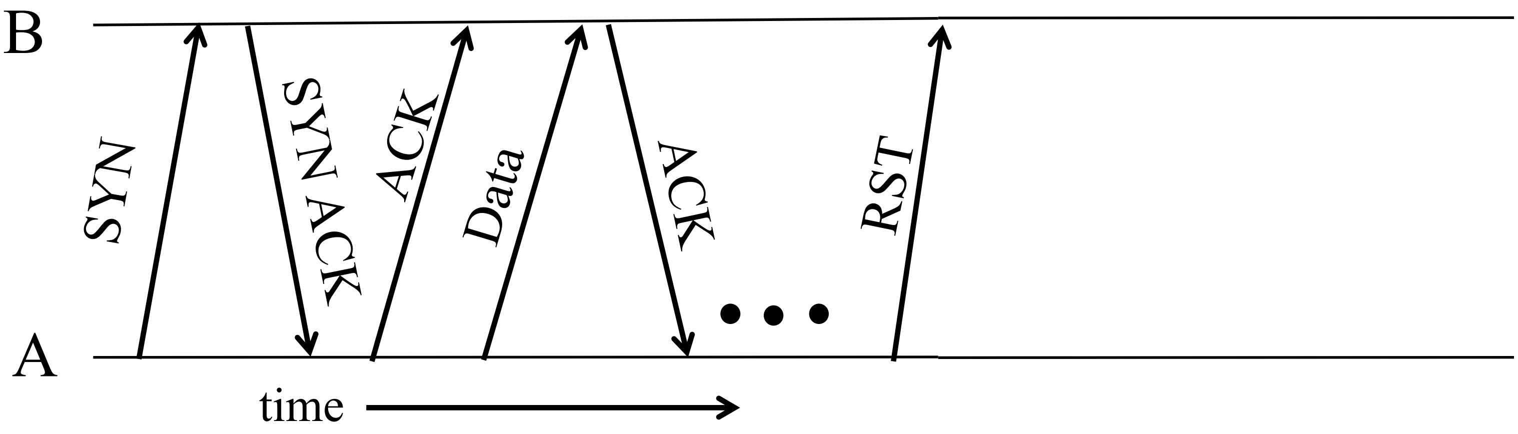
Checksum

Urgent pointer

Options (variable)

Data

Abrupt Termination

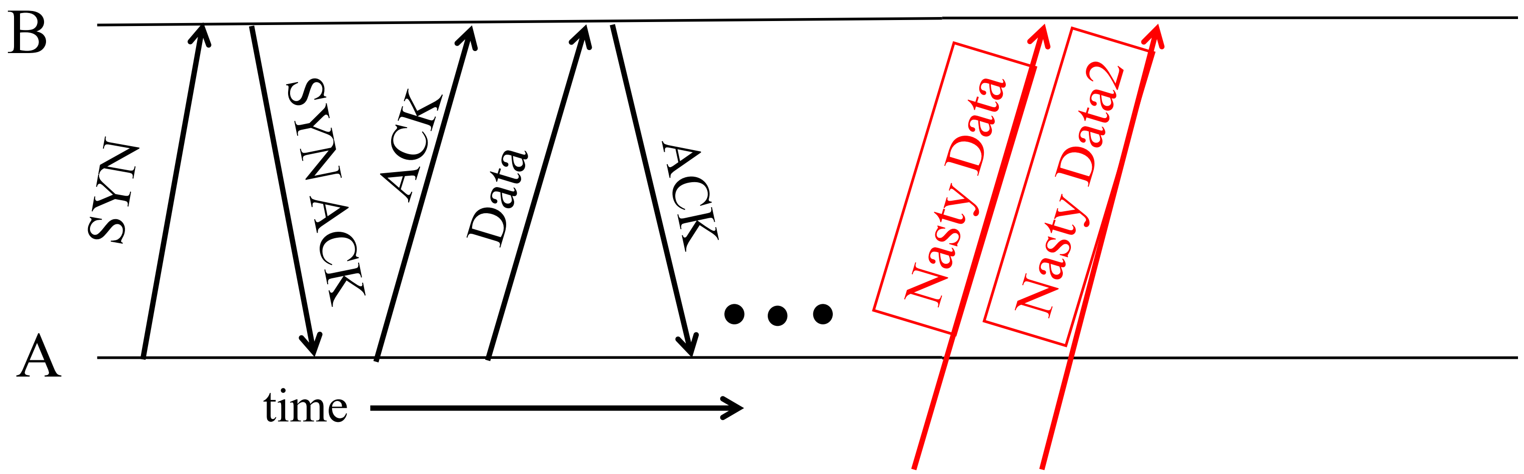


- A sends a TCP packet with RESET (**RST**) flag to B
  - E.g., because application process on A crashed
- Assuming that the sequence numbers in the **RST** fit with what B expects
  - No further communication on connection is possible

TCP Threat: Disruption

- Normally, TCP finishes ("closes") a connection by each side sending a `FIN` control message
  - Reliably delivered, since other side must ack
- But: if a TCP endpoint finds unable to continue (process dies; info from other "peer" is inconsistent), it abruptly terminates by sending a `RST` control message
  - Unilateral
  - Takes effect immediately (no ack needed)
  - Only accepted by peer if has correct sequence number
- So: if attacker knows ports & sequence numbers, can disrupt any TCP connection

TCP Threat: Injection



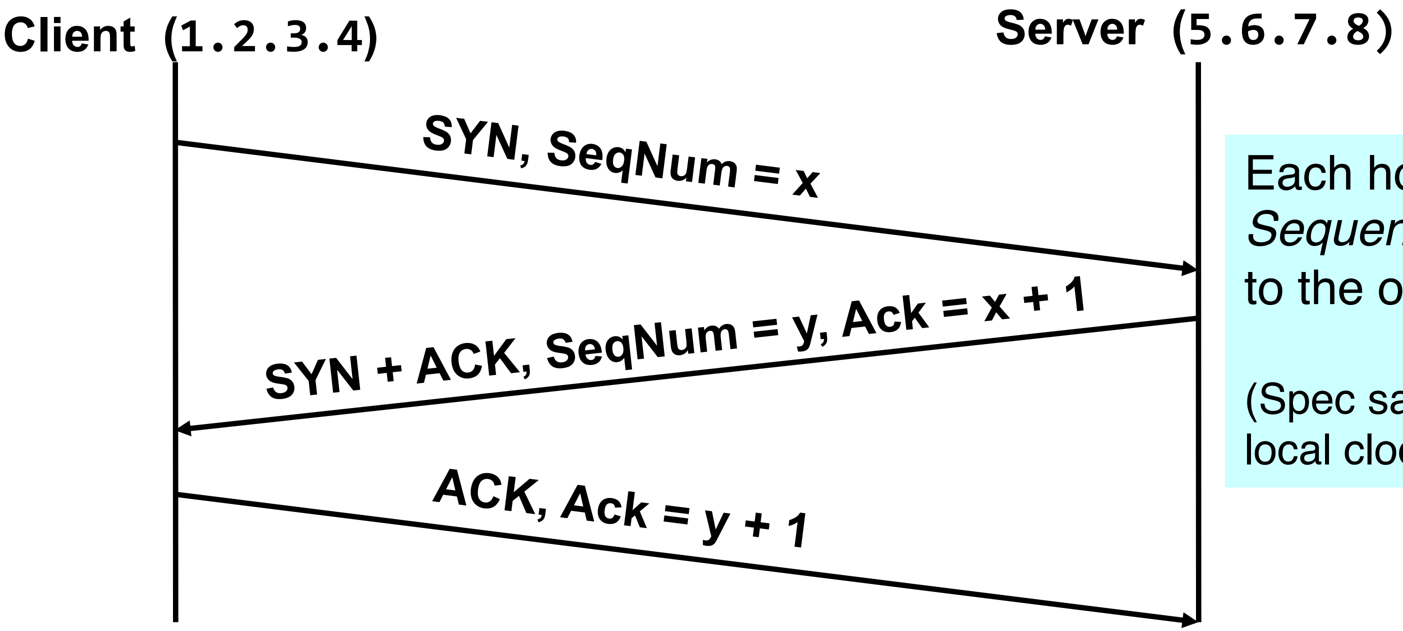
- What about inserting data rather than disrupting a connection?
  - Again, all that's required is attacker knows correct ports, seq. numbers
- Termed TCP connection hijacking (or "*session hijacking*")
  - General means to take over an already-established connection!
- If an attacker can see our TCP traffic?
  - Then they immediately know the port & sequence numbers
- If not, guess the port & sequence numbers

TCP Threat: Injection via Spoofing

- Create a fake connection, rather than inject into a real one
  - Why?
  - Leverage a server's trust of a given client as identified by its IP address
  - The attacker can't be traced back

TCP Threat: Spoofing

- TCP connection establishment:



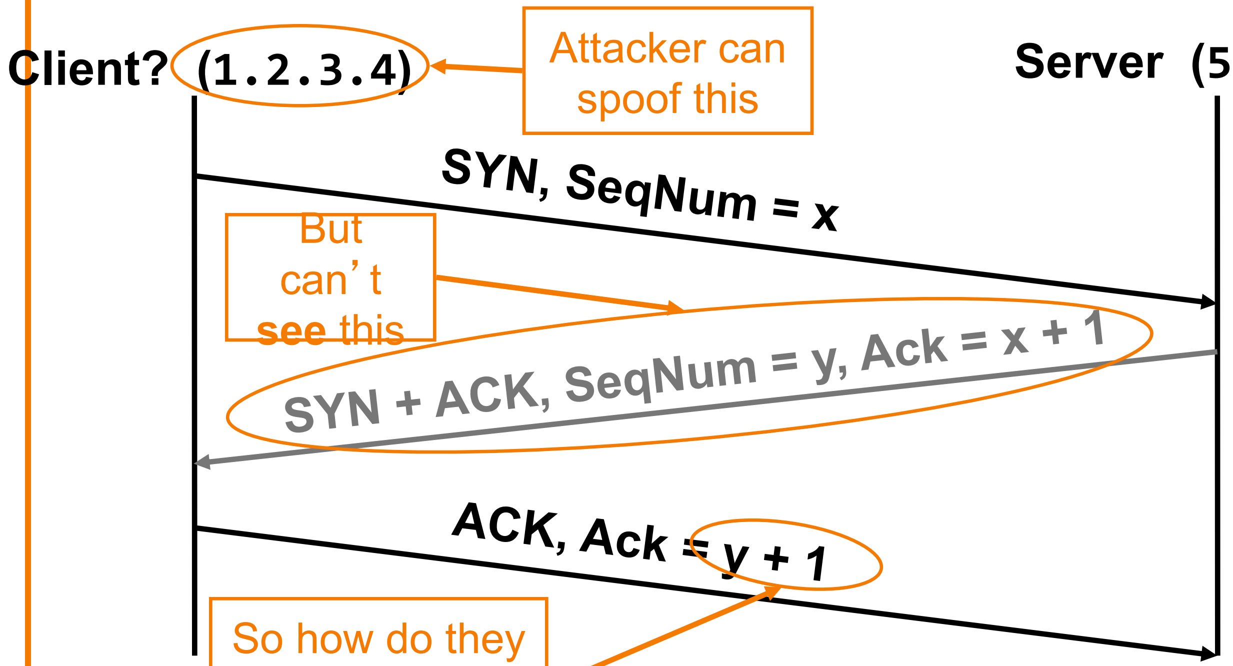
Each host tells its *Initial Sequence Number* (ISN) to the other host.

(Spec says to pick based on local clock)

- How can an attacker create an *apparent but fake* connection from `1.2.3.4` to `5.6.7.8`?

Spoofing: Attacker’s Viewpoint

Attacker



Attacker can spoof this

But can’t **see** this

So how do they know what to put here?

Each host tells its *Initial Sequence Number* (ISN) to the other host.

(Spec says to pick based on local clock)

Hmm, any way for the attacker to know ***this?***

How Do We Fix This?

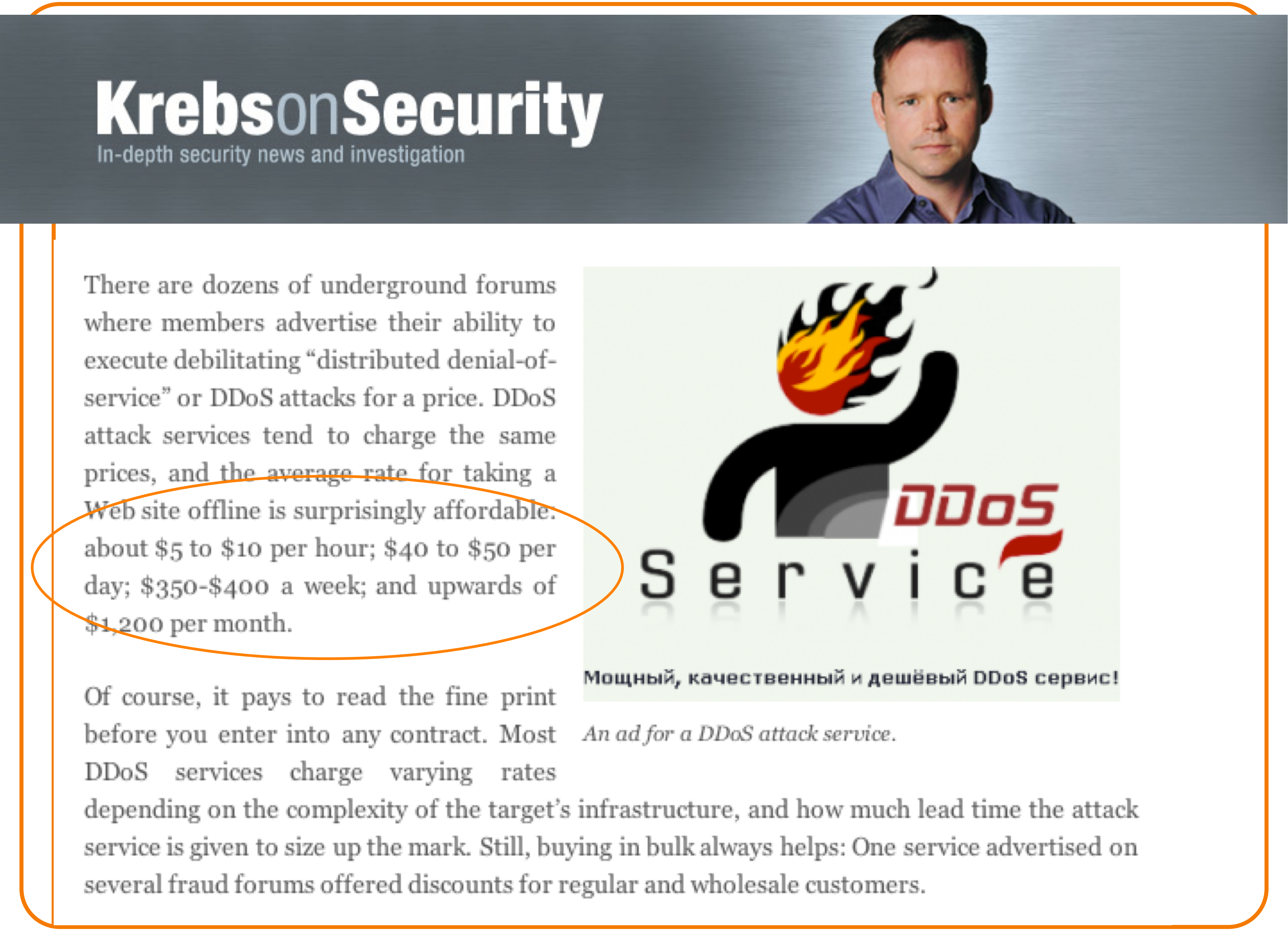
Use A *Random* ISN

Sure - make a non-spoofed connection *first*, and see what server used for ISN y then!

Denial-of-Service (DoS) Attacks

Attacks on Availability

- Denial-of-Service (DoS)
- Preventing legitimate users from using a service
- DDoS: Distributed Denial-of-Service
  - Attacks from multiple hosts on the Internet
- We need to consider our threat model
  - What might motivate a DoS attack?



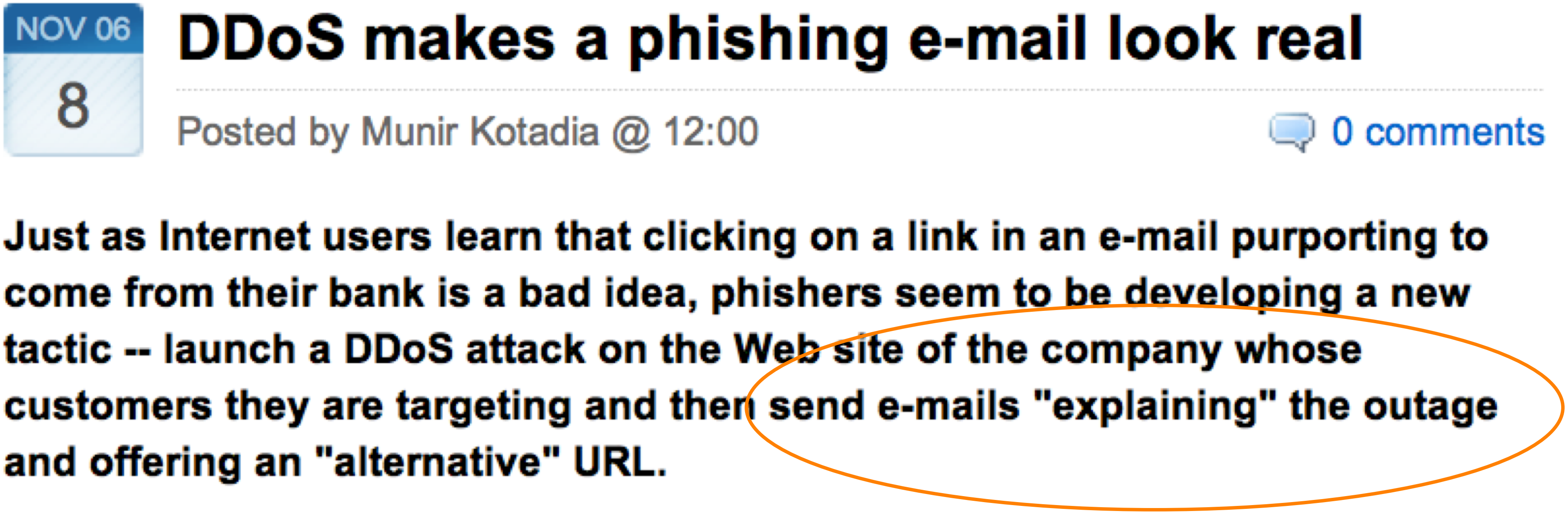
KrebsonSecurity

In-depth security news and investigation

There are dozens of underground forums where members advertise their ability to execute debilitating "distributed denial-of-service" or DDoS attacks for a price. DDoS attack services tend to charge the same prices, and the average rate for taking a Web site offline is surprisingly affordable: about $5 to $10 per hour; $40 to $50 per day; $350-$400 a week; and upwards of $1,200 per month.

An ad for a DDoS attack service.

Of course, it pays to read the fine print before you enter into any contract. Most DDoS services charge varying rates depending on the complexity of the target's infrastructure, and how much lead time the attack service is given to size up the mark. Still, buying in bulk always helps: One service advertised on several fraud forums offered discounts for regular and wholesale customers.



NOV 06
8

DDoS makes a phishing e-mail look real

Posted by Munir Kotadia @ 12:00

Just as Internet users learn that clicking on a link in an e-mail purporting to come from their bank is a bad idea, phishers seem to be developing a new tactic -- launch a DDoS attack on the Web site of the company whose customers they are targeting and then send e-mails "explaining" the outage and offering an "alternative" URL.

Motivations for DoS

- Showing off / entertainment / ego
- Competitive advantage
  - Maybe commercial, maybe just to win
- Economic benefits
- Political statements
- Cyber warfare

Most Significant Operational Threats

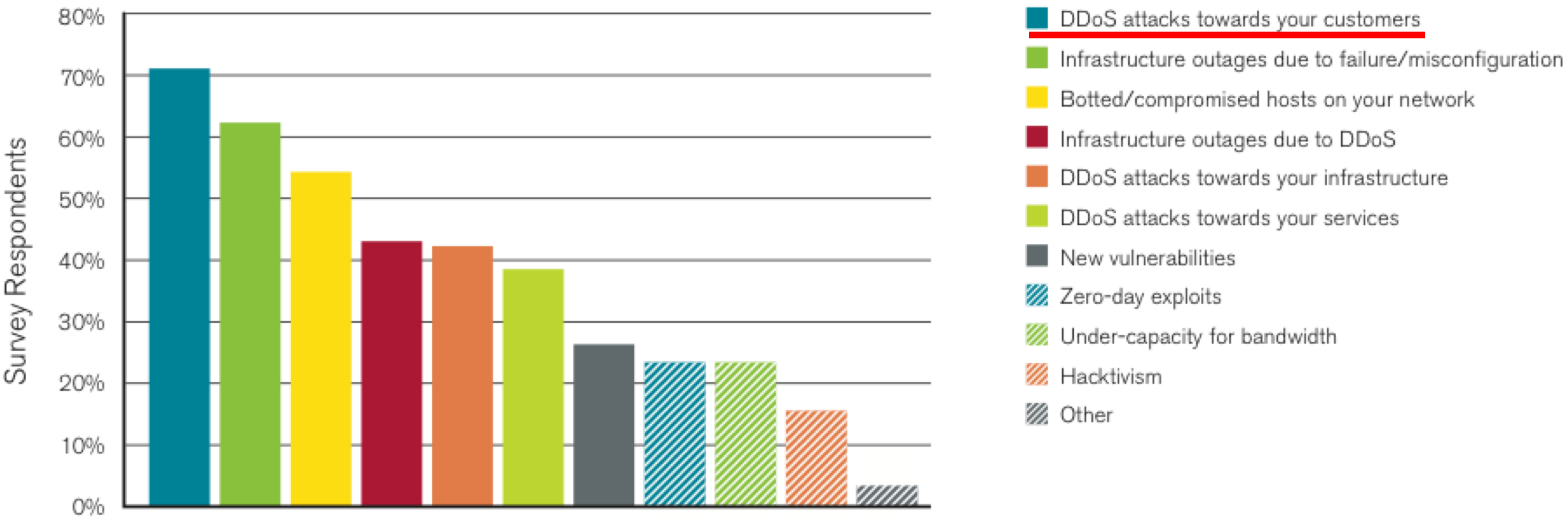
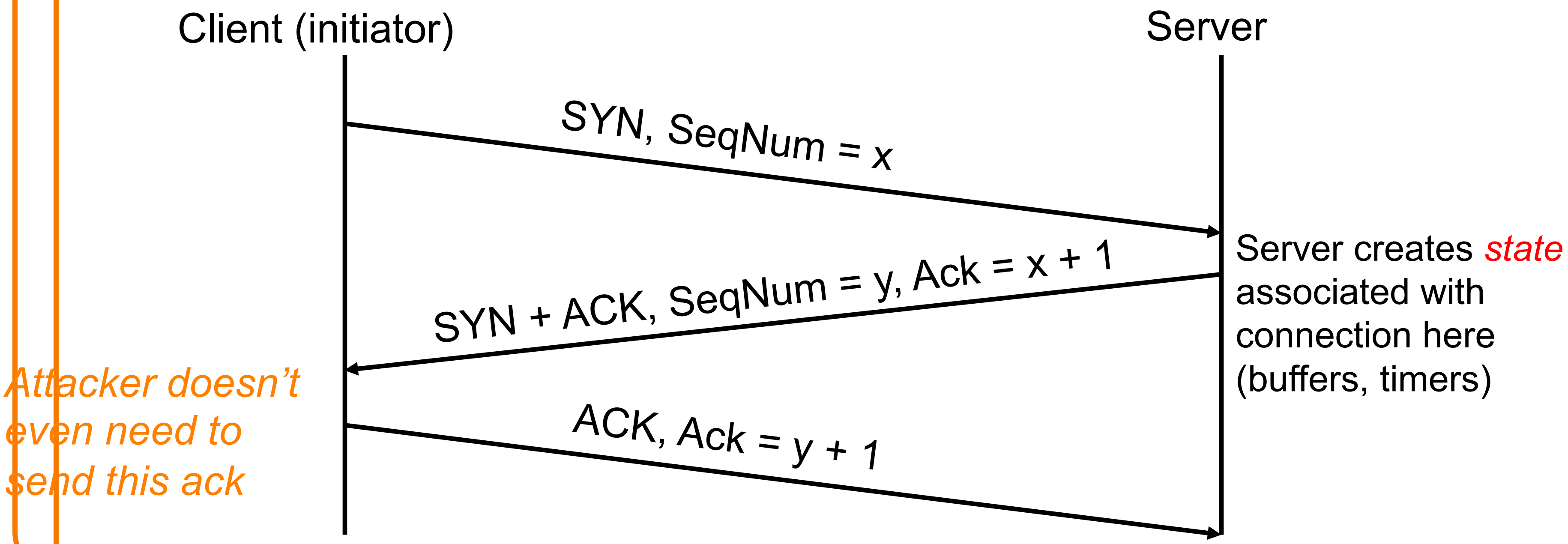


Figure 6 *Source: Arbor Networks, Inc.*

Transport-Level Denial-of-Service

- Recall TCP's 3-way connection establishment handshake
  - Goal: agree on initial sequence numbers

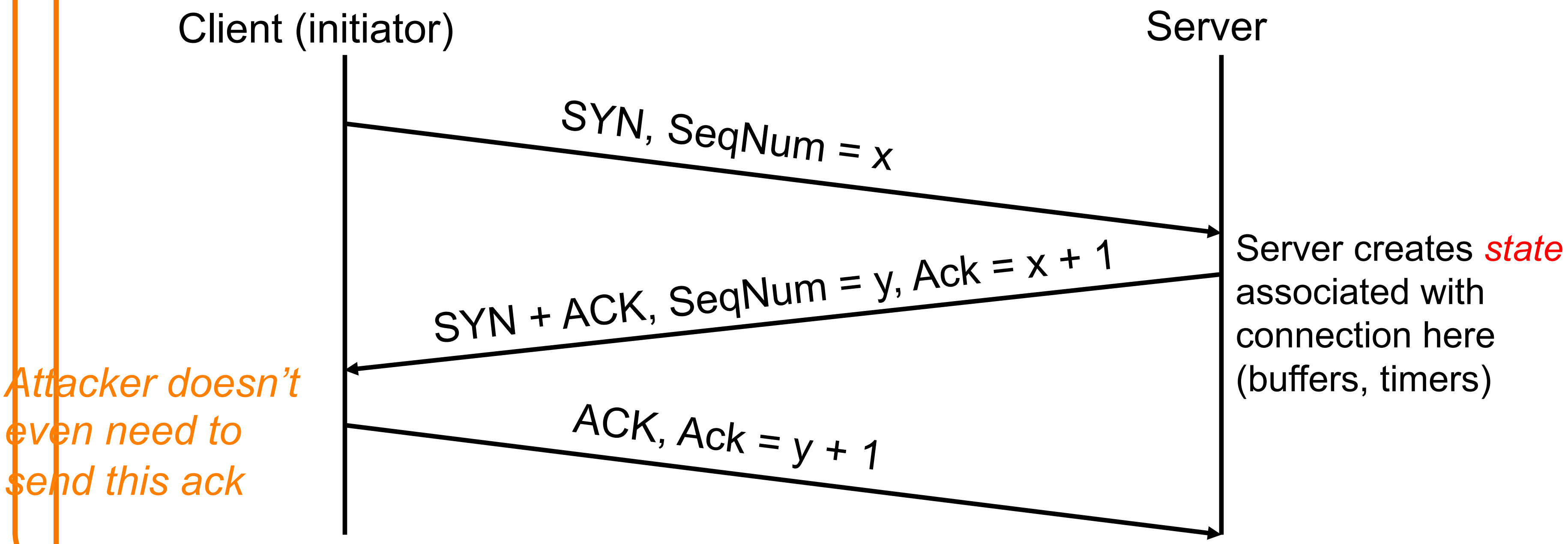


Server creates *state* associated with connection here (buffers, timers)

Attacker doesn't even need to send this ack

Transport-Level Denial-of-Service

- Recall TCP's 3-way connection establishment handshake
  - Goal: agree on initial sequence numbers
- So a single SYN from an attacker suffices to force the server to spend some memory



Server creates *state* associated with connection here (buffers, timers)

Attacker doesn't even need to send this ack

TCP *SYN Flooding*

- Attacker targets memory of the server
- Every (unique) SYN that the attacker sends burdens the target
- What should target do when it has no more memory for a new connection?
  - No good answer
  - Refuse new connection?
    - Legit new users can’t access service
  - Evict old connections to make room?
    - Legit old users get kicked off

TCP SYN Flooding Defense

- How can the target defend itself?

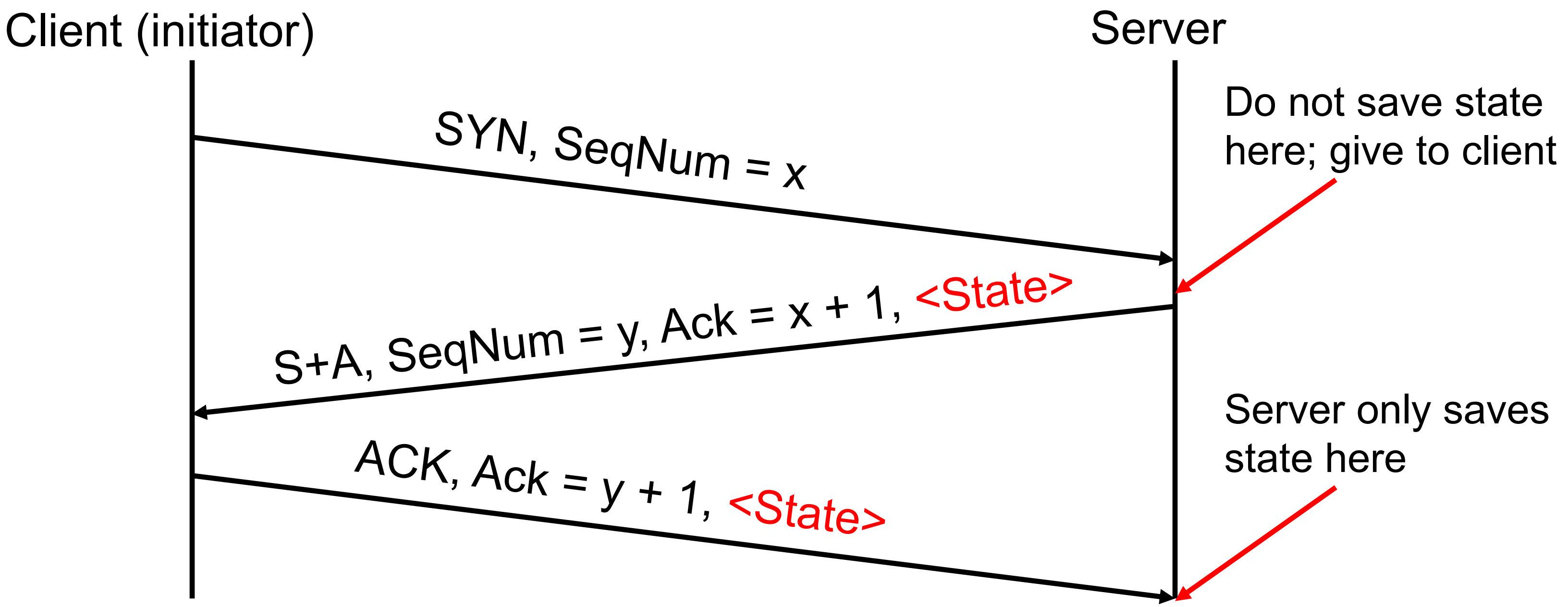
- Approach #1: tons of memory
  - How much is enough?
  - Depends on resources attacker can bring to bear, which might be hard to know

TCP SYN Flooding Defense

- Approach #2: identify bad actors & refuse connections
  - Hard because identification is on IP address
  - For a public Internet service, who knows which addresses customers might come from?
  - Plus: attacker can spoof addresses since they don’t need to complete TCP 3-way handshake
- Approach #3: don’t keep state!
  - “SYN cookies”; only works for spoofed SYN flooding
  - Attacker can use botnet to launch DDoS

SYN Flooding Defense: Idealized

- Server: when SYN arrives, rather than keeping state locally, send it to the client …
- Client needs to return the state in order to establish connection


Client (initiator)
Server
SYN, SeqNum = x
S+A, SeqNum = y, Ack = x + 1, <State>
ACK, Ack = y + 1, <State>
Do not save state here; give to client
Server only saves state here

SYN Flooding Defense: Idealized

- Server: when SYN arrives, rather than keeping state locally, send it to the client …
- Client
  establ

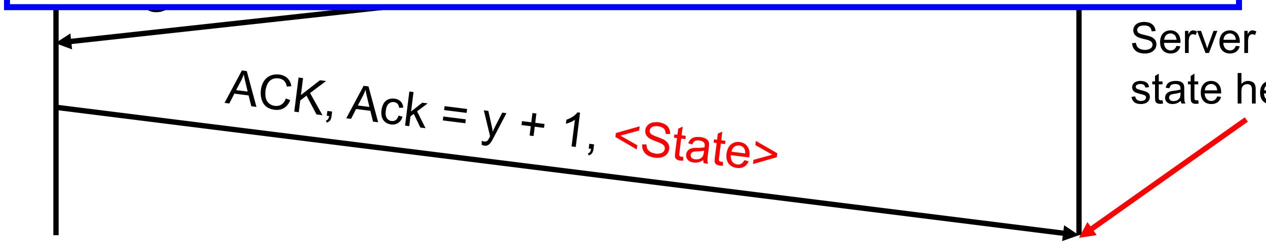
Problem: the world isn't so ideal!

TCP doesn't include an easy way to add a new **<State>** field like this.

Is there any way to get the same functionality without having to change TCP?

Client (

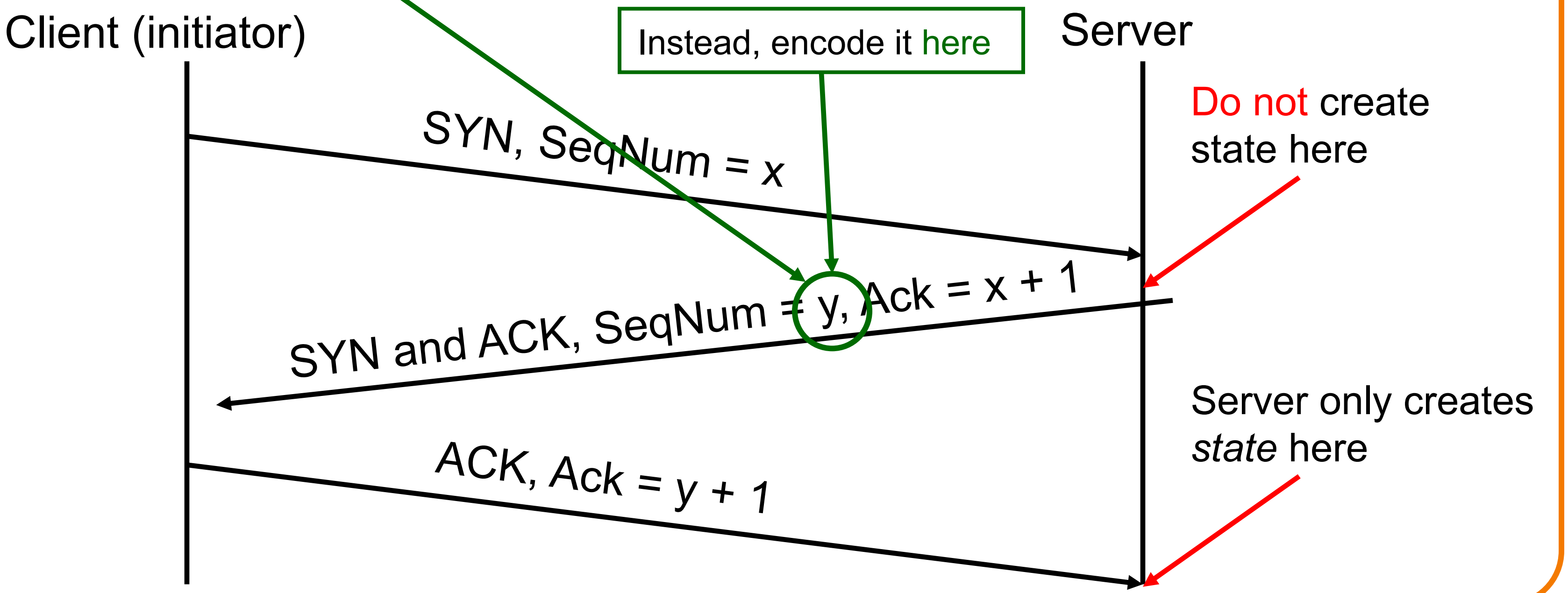
t save state
give to client



Server only saves state here

Practical Defense: SYN Cookies

- Server: when SYN arrives, encode connection state entirely within SYN-ACK's sequence # y
  - y = encoding of necessary state
- When ACK of SYN-ACK arrives, server creates state



Client (initiator)

Server

Instead, encode it here

Do not create state here

SYN, SeqNum = x

SYN and ACK, SeqNum = y, Ack = x + 1

ACK, Ack = y + 1

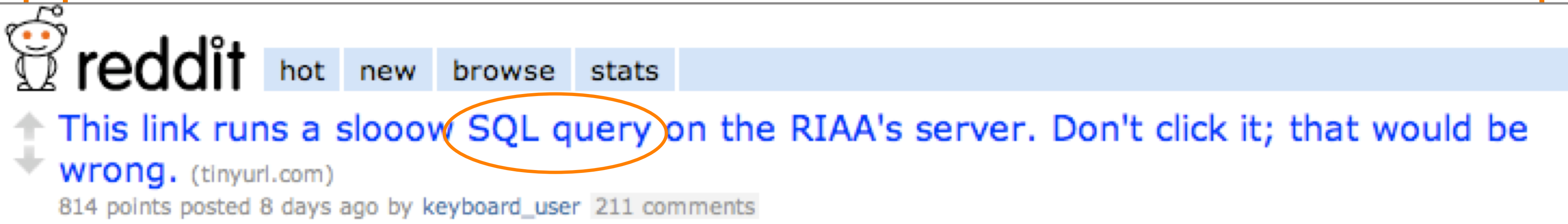
Server only creates *state* here

SYN Cookies: Discussion

- Illustrates general strategy: rather than holding state, encode it so that it is returned when needed
- For SYN cookies, attacker must complete 3-way handshake in order to burden server
  - Can’t use spoofed source addresses
- Note #1: strategy requires that you have enough bits to encode all the state
- Note #2: if it’s expensive to generate or check the cookie, then it’s not a win

Application-Layer DoS

- Rather than exhausting memory resources, attacker can overwhelm a service’s processing capacity
- There are many ways to do so, often at little expense to attacker compared to target (asymmetry)



This link runs a slooow SQL query on the RIAA's server. Don't click it; that would be wrong. (tinyurl.com)

814 points posted 8 days ago by keyboard_user 211 comments

The link sends a request to the web server that requires heavy processing by its “backend database”.

Algorithmic complexity attacks

- Attacker can try to trigger worst-case complexity of algorithms / data structures
- Example: You have a hash table.
  Expected time: $O(1)$ Worst-case: $O(n)$
- Attacker picks inputs that cause hash collisions.
  Time per lookup: $O(n)$
  Total time to do n operations: $O(n^2)$
- Solution? Use algorithms with good worst-case running time.

Summary

- TCP disruption
- TCP injection
- TCP spoofing
- SYN flooding
  - DoS