Denial-of-Service / Wireless / Multimedia

EE 122: Intro to Communication Networks
Fall 2006 (MW 4-5:30 in Donner 155)

Vern Paxson
TAs: Dilip Antony Joseph and Sukun Kim
http://inst.eecs.berkeley.edu/~ee122/
Materials with thanks to Jennifer Rexford, Ion Stoica, and colleagues at Princeton and UC Berkeley

Announcements

• Office hours this week by appointment
• I’ll be giving a lecture on Experiences With Countering Internet Attacks next Wednesday, 2:30-4PM in Cory 540 A/B (optional!)
• Next Lecture: Final Review
  – and course evaluation

Goals of Today’s Lecture

• Denial-of-Service
  – Transport layer (SYN flooding)
  – Application layer (CAPTCHAs)
• Wireless link layers
  – 802.X, Bluetooth
• Issues for transmitting multimedia content
  – Audio
  – Video
  – Voice-over-IP (VOIP)

Recap: Defending Against Network Flooding

• How do we defend against such floods?
• Answer: basically, we don’t! Big problem today!
• Techniques exist to trace spoofed traffic back to origins, but this isn’t useful in face of a large attack
• Techniques exist to filter traffic, but a well-designed flooding stream defies stateless filtering
• Best solutions to date:
  – Overprovision - have enough raw capacity that it’s hard to flood your links
    • Largest confirmed botnet to date: 1.5 million hosts
  – Distribute your services - force attacker to flood many points
    • E.g., the root name servers

Transport-Level Denial-of-Service

• Recall TCP’s 3-way connection establishment handshake
  – Goal: agree on initial sequence numbers
  – Starting sequence numbers are based on clock
  – Random

  to prevent attacker from guessing them to establish connections using spoofed source addresses

  Client (initiator)

  SYN, SeqNum = x

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

  ACK, Ack = y + 1

  Server

  Server creates state associated with connection here

  Server

  Server

SYN Flooding

• Attacker sends victim TCP SYNs with random client ports and spoofed source address
• Victim responses with SYN+ACKs
  – Victim also allocates memory for connection, sets timers
  – Holds memory until 3-way handshake completes
  – Or until eventual timeout (e.g., 3 minutes)
• Victim quickly runs out of memory
  – Newly arriving connections are denied
  – Many of these are the attacker’s bogus conn. attempts
  – But others are legitimate. No one new can get to the site.
• Note: network capacity/overprovisioning doesn’t help
Flooding Defense: SYN Cookies

- Server, when SYN arrives, encode connection state entirely within SYN-ACK's sequence # y
  \[ y = \text{SHA-1} (\text{client_addr}, \text{client_port}, \text{ISN} x, \text{server_secret}) \]
- When ACK of SYN-ACK arrives, server only creates state if seq # y in it agrees with hash

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 the server
  - Can’t use spoofed source addresses
- Note #1: strategy requires that you have enough bits to encode all the state
  - This is just barely the case for SYN cookies
  - You only have 32 bits to work with in server’s ISN
  - (And not the case once connection is established)
- Note #2: if it’s expensive to generate or check the cookie, then it’s not a win

Application-Layer Flooding

- Attacker makes a lot of expensive service requests
  - E.g., http://victim.com/back_end_database.cgi?search...
- Expense of request gives attacker leverage
  - It can also be very hard to tell legitimate requests from bogus
- CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart)
  - Idea: “Reverse Turing Test”
  - Prove that a client is a human rather than a machine
  - Based on known-hard AI problems that humans solve readily
  - Drawbacks:
    - If visual, discriminates against blind users
    - Sometimes you want machines to be able to make (legit) requests
    - Depending on the problem, an arms race (driving technology forward)

Summary of Denial-of-Service

- Can occur at different semantic levels
  - Network layer vs. transport layer vs. application layer
  - Very hard to address if attacker has a lot of zombies
- Principle: attacker finds bottleneck element …
  - ... and sends it more work than it can cope with
  - E.g.: Router’s packets-per-second processing capability
  - Link’s bits-per-second transmission capability
  - End host’s memory available for new connections ...
  - ... or cycles available to validate connections (cookies)
  - Server’s cycles for processing requests
- Defend via
  - Overprovisioning
  - Force sender to prove they’re not spoofing (cookies)
  - Force sender to prove they’re not a robot (CAPTCHAs)
Wireless Media Access (courtesy of S. Savage, UCSD)

- Wireless links are extremely convenient
- What makes wireless links more problematic than wired links?
  - Why not just use Ethernet algorithms?
- It’s technically difficult to detect collisions
  - Transmitter swamps co-located receiver
- … even if we could, it wouldn’t work
  - Different transmitters have different coverage areas
- In addition, wireless links are much more prone to loss than wired links

Hidden Terminals

A - B - C

- A and C can both send to B but can’t hear each other
  - A is a hidden terminal for C and vice versa
- CSMA/CD will be ineffective – need to sense at receiver

Exposed Terminals

A - B - C - D

- B, C can hear each other …
- … But can safely send to A, D

CSMA/CA: CSMA w/ Collision Avoidance

- Since we can’t detect collisions, we try to avoid them
- When medium busy, choose random interval (contention window)
  - Wait for that many idle timeslots to pass before sending
- When a collision is inferred, retransmit with binary exponential backoff (like Ethernet)
  - Use ACK from receiver to infer “no collision”
  - Use exponential backoff to adapt contention window

RTS / CTS Protocols (MACA)

MACA = Multiple Access with Collision Avoidance

Overcome exposed/hidden terminal problems with contention-free protocol
1. B stimulates C with Request To Send (RTS)
2. A hears RTS and defers (to allow C to answer)
3. C replies to B with Clear To Send (CTS)
4. D hears CTS and defers to allow the data
5. B sends to C

MACA, con’t

- If sender doesn’t get a CTS or ACK back, it assumes collision
- If other nodes hear RTS, but not CTS: send
  - Presumably, destination for sender is out of node’s range …
  - … Can cause problems when a CTS is lost
IEEE 802.11 Wireless LAN ... etc.

- **802.11b**
  - 2.4-5 GHz unlicensed radio spectrum
  - E.g., microwave ovens, cordless phones
  - up to 11 Mbps
  - direct sequence spread spectrum (DSSS) in physical layer
  - All hosts use same code
  - Widely deployed, using base stations
    - Base station provides gateway from wireless nodes to another hop
    - Next hop could be another base station ... or could be the Internet

- **802.11a**
  - 5-6 GHz range
  - up to 54 Mbps

- **802.11g**
  - 2.4-5 GHz range
  - up to 54 Mbps

- **Bluetooth**
  - Separate (non-802) standard for short-range wireless
  - Primarily meant for connecting nearby devices (e.g., keyboards/mice)
  - Enabled for many cell phones / laptops
  - ‘Discoverable’ ...

- **802.11n**
  - 2.4-5 GHz range
  - up to 54 Mbps

Wireless: Our Security Headaches Grow

Our Security Headaches Grow, con’t
5 Minute Break

Questions Before We Proceed?
Multimedia

Digital Audio

- **Capturing** the analog signal
  - Sample at some fixed rate
  - Each sample is an arbitrary real number
- **Quantizing** each sample
  - Round each sample to one of a finite number of values
  - Represent each sample in a fixed number of bits

4 bit representation
(values 0-15)

Audio Examples

- **Speech**
  - Sampling rate: 8000 samples/second
  - Sample size: 8 bits per sample
  - Rate: 64 kbps

- **Compact Disc (CD)**
  - Sampling rate: 44,100 samples/second
  - Sample size: 16 bits per sample
  - Rate: 705.6 kbps for mono,
    1.411 Mbps for stereo

Reduce these via **compression**… (remove redundancy)

Digital Video

- **Capturing** the analog signal
  - Sample at some fixed rate (e.g., 24 or 30 times per sec)
  - Each sample is an image
- **Quantizing** each sample
  - Representing images as array of picture elements
  - Each pixel is a mixture of colors (red, green, and blue)
  - E.g., 24 bits, with 8 bits per color

Video Compression: Within an Image

- **Image compression**
  - Exploit spatial redundancy (e.g., regions of same color)
  - Exploit aspects humans tend not to notice
- **Common image compression formats**
  - Joint Photographic Experts Group (JPEG)
  - Graphical Interchange Format (GIF)

Uncompressed: 167 KB Good quality: 46 KB Poor quality: 9 KB
Video Compression: Across Images

- Exploit temporal redundancy across images
- Common video compression formats
  - MPEG 1: CD-ROM quality video (1.5 Mbps)
  - MPEG 2: high-quality DVD video (3-6 Mbps)
  - Proprietary protocols like QuickTime and RealNetworks

Transferring Audio and Video Data

- Simplest case: just like any other file
  - For off-line listening/viewing
- Harder: someone listening/viewing in real time
  - Smooth "playback" required or else gaps/glitches immediately perceived
  - Data must arrive quickly enough or else receiver stalls
- Harder still: two (or more) people interacting in real time
  - Delays over 150-200 msec very noticeable
  - ... and delays over 400 msec a disaster
  - Clearly, can only go so far without network support
  - E.g., telephone or real-time game

Playout Buffer

- Client buffer
  - Store the data as it arrives from the server
  - Play data for the user in a continuous fashion
- Playout delay
  - Client waits a bit before playing
  - ... to allow some data to build up in the buffer
  - ... to help tolerate some delays down the road

Playout Buffer/Delay, con’t

Summary of Playout Choices

- Play back data upon arrival
  - Distorted signal
- Buffer data for a while (playback buffer)
  - Extra delay, less distortion
  - Absorb jitter (variation in delay)
- Tradeoff depends on application (and use)
  - Noninteractive: absorb delay, eliminate all distortion
  - Interactive: absorb only a little delay, eliminate some distortion
- More generally, compression + playout delay allow application to adapt to network conditions
  - Graceful degradation - you at least get something

Voice Over IP (VoIP)

- Delivering phone calls over IP
  - Computer to computer
  - Analog phone to/from computer
  - Analog phone to analog phone
- Motivations for VoIP
  - Cost reduction
  - Simplicity
  - Advanced applications
    - Web-enabled call centers
    - Collaborative white boarding
    - Do Not Disturb, Locate Me, etc.
    - Voicemail sent as e-mail
**Summary of Multimedia**

- Data rates can be significant
  - Leads to emphasis on compression
- For real-time or interactive reception, must accommodate vagaries of variable delay (jitter)
  - Playback buffer: hold data up to maximum limit to enable smooth playback
- Voice-over-IP: convergence of global telephony network with global data network