Countering DDoS Without Changing The Internet

Angelos Keromytis
Columbia University
angelos@cs.columbia.edu
Overview

• Provide service to a DoS’ed site without ISP support
  › Or at least not much...
  › ISPs don’t care or don’t want to spend money

• Use filtering at near-edge router
  › Use characteristics easy to filter in hardware

• Obscure "secret" needed to get through filtering
  › Use overlay to route traffic to secret location

• Authenticate clients
• Joint work with Vishal Misra and Dan Rubenstein
• Support from DARPA, Cisco, Intel
Distributed Denial of Service (DDoS)

Potential congestion points

Computation DoS
Approach

• Separate good from bad/unknown traffic
  › Authenticate users for entering the overlay
  › Route good traffic through overlay

• Treat good traffic preferentially
  › Filter on packet characteristic
    • Routers can filter source IP address VERY fast
  › Vary characteristic with time

• Attacker must guess, or attack infrastructure
Overview of SOS

- Beacon
- Servlet
- Secret
- Overlay nodes
- SOAP
- Source Point
- Target
- Filtered region
Assumptions

- Attackers cannot saturate Internet core
- Routers can filter efficiently on specific header fields
  - Source IP address
  - GRE Key field ?
- Attackers cannot eavesdrop on arbitrary links
- Well-known users
  - No assumptions about location, IP address, etc.
- Minimal cooperation with home ISP
  - But read on...
Architecture details

• Crypto-based access control
  ▶ Standard security protocols (IPsec, TLS, ...)
  ▶ Certificates to authenticate users -> stateless SOAPs
  ▶ Requires notion of authenticated user

• Chord-based overlay
  ▶ Self-healing

• Target changes Secret Servlet periodically...
  ▶ ...or as attacks are detected
More details

• Overlay can be deployed at end-nodes
  ▶ Collaboratively?
• Can be used concurrently by multiple entities
• Defense on demand
• Use-based charging or premium service
Experimental prototype

- Implementation for the web (WebSOS)
- Use SSL to protect traffic inside overlay
- Use SSL to authenticate user to overlay
- Unmodified browsers and web servers
  - Java applet on browser for first-hop SSL encapsulation and SOAP selection
- Overlay nodes implemented as proxies
- Deployed over PlanetLab
Experimental results

- Use overlay only to determine secret servlet
  - SOAP queries Beacon
  - Cache replies, periodically refresh
- Route data directly to secret servlet

<table>
<thead>
<tr>
<th>Server</th>
<th>Direct</th>
<th>Original Request</th>
<th>Cached Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yahoo!</td>
<td>1.39</td>
<td>4.15</td>
<td>3.67</td>
</tr>
<tr>
<td>Verisign</td>
<td>3.43</td>
<td>7.33</td>
<td>6.77</td>
</tr>
<tr>
<td>Columbia BB</td>
<td>0.64</td>
<td>3.97</td>
<td>3.43</td>
</tr>
<tr>
<td>Columbia BB (2nd)</td>
<td>0.14</td>
<td>0.55</td>
<td>0.56</td>
</tr>
</tbody>
</table>

- Recovery from attacks in under 10 seconds
Remaining issue

- Requiring known users is too restrictive
- What we really want is guarantee no "zombies"
- Graphic Turing Tests
  - Tests that humans can perform, but difficult for computers
Graphic Turing Tests

CAPTCHA Implementation for SOS Project

Please type the text you see in the above picture

[Text Field]  Submit  Reset

please hit refresh/reload to view another image

This Captcha library was obtained from CMU CAPTCHA Project
WebSOS with GTT
Migrating OVErlay (MOVE)
Operation
MOVE Performance

- Overlay performance same as WebSOS
- Service disruption to user: time it takes to migrate server

<table>
<thead>
<tr>
<th>Migration server</th>
<th>Network Latency (round-trip)</th>
<th>Migration Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-located (using NFS)</td>
<td>1.02ms</td>
<td>0.761s</td>
</tr>
<tr>
<td>Co-located (using SCP)</td>
<td>1.02ms</td>
<td>1.162s</td>
</tr>
<tr>
<td>Remote (using SCP)</td>
<td>10.6ms</td>
<td>6.632s</td>
</tr>
</tbody>
</table>
Pointers

• Papers
  ▶ SIGCOMM 2002
    • http://www.cs.columbia.edu/~angelos/Papers/sos.pdf
  ▶ ICON 2003
    • http://www.cs.columbia.edu/~angelos/Papers/websos-icon.pdf
  ▶ CCS 2003
    • http://www.cs.columbia.edu/~angelos/Papers/websos.pdf

• Questions?
  ▶ angelos@cs.columbia.edu