Using Graphic Turing Tests to Counter Automated DDoS Attacks Against Web Servers

Angelos Stavrou

joint work with:
William G. Morein, Debra L. Cook
Angelos D. Keromytis, Vishal Misra, Dan Rubenstein
Columbia University, Computer Science Department
Overview of the talk

• Description of the Problem
  - What are DDoS Attacks?

• Description of the SOS Architecture
  - Previous Work

• Our Approach (WebSOS)
  - Extensions to SOS

• Experimental Results

• Other Approaches

• Future Work
Network DoS

• Over a network
  ▶ No need to be a legitimate user

• Action at a distance
  ▶ Minimize risk of exposure

• Easily Automated
  ▶ Can use “hijacked” machines
  ▶ Distributed DoS (DDoS)
DDoS Attack
Network with SOS

User

ISP

Access point

Access point

Access point

Access point

Access point

Secret Servlet

Target Server

Overlay Nodes

Router Filtered Area
DDoS in an SOS-Equipped network

![Diagram showing the process of DDoS in an SOS-Equipped network](image)

- **Attacker** and **User** access the system through **ISP**.
- **Overlay Nodes** are connected to the **Target Server**.
- **Router Filtered Area** prevents incoming traffic from being blocked.
DDoS in an SOS-Equipped network

Overlay nodes
Our 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
Remaining issue

• Prevent Large Scale Automated Attacks, allow enough time for the overlay system to “heal”

• Requiring known users is too restrictive

• What we really want is guarantee no "zombies"
Solution

• Extend SOS with 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

Submit  Reset

please hit refresh/reload to view another image

This Captcha library was obtained from CMU CAPTCHA Project
WebSOS with GTT

Session Request
Experimental prototype

• Implementation for web services (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
• Overlay nodes implemented as proxies
• Deployed over PlanetLab
Experimental results

Latency Overhead

Time (in seconds)

Number of nodes participating in the Overlay network

- Yahoo!
- Verisign
- Columbia BB
- Columbia BB (2nd)
Another Approach: Short-cut Routing

- Use overlay only to determine secret servlet
- Route data from the Access Point directly to secret servlet
Other Approaches

• Pushback
  • Support from the routers
  • Trust issues

• Probabilistic Packet Marking (PPM)
  • Support from the routers

• Polling-based Traceback
  • Hardware support required
Open Problems, Future Directions

• Deploy WebSOS in a large scale network
• Use WebSOS protection for services other than Web
• Detect and prevent attacks from within WebSOS