Kill-Bots: Surviving DDoS Attacks That Mimic Legitimate Browsing

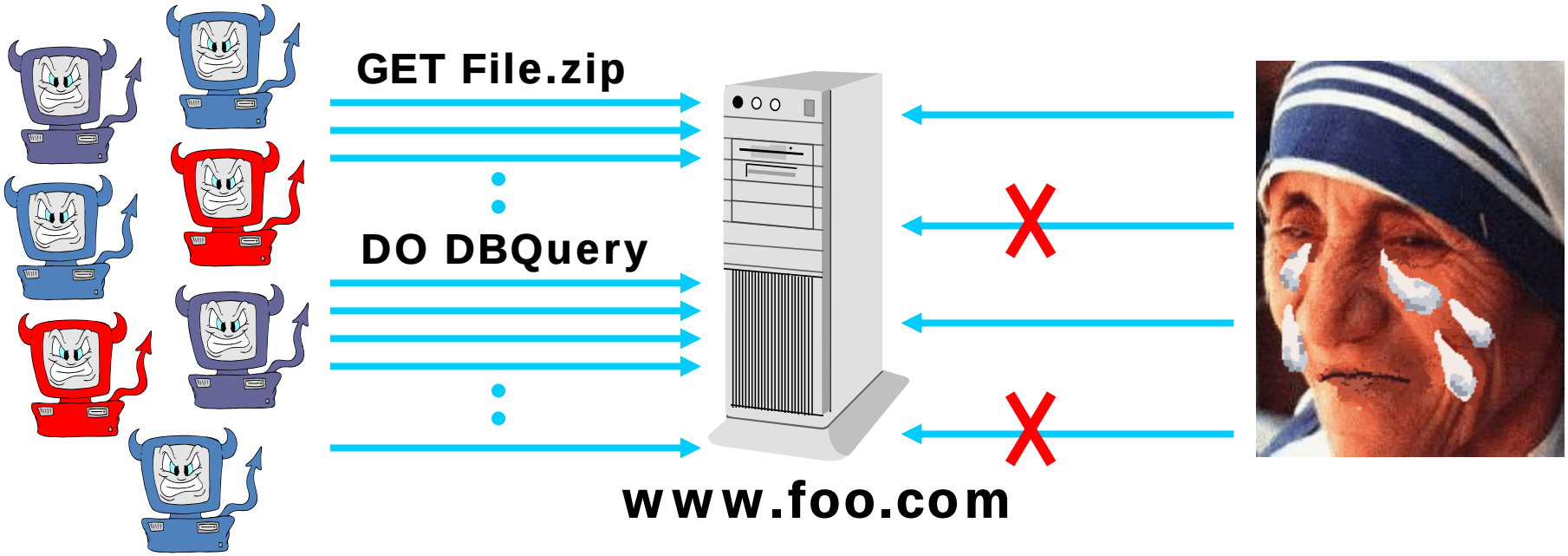
Srikanth Kandula

Dina Katabi, Matthias Jacob, and Arthur Berger

CyberSlam = DDoS that Mimics Legitimate Browsing

CyberSlam

20,000+ zombies issue requests that mimic legitimate browsing



Requests Look Legitimate ⇒ Standard filters don't help

CyberSlam Attacks Happen!

- Instances of CyberSlam
  - First FBI DDoS Case – Hired professionals hit competitor
  - Mafia extorts online gaming sites …
  - Code RED Worm
- Why CyberSlam?
  - Avoid detection by NIDS & firewalls
  - High pay-off by targeting expensive resources
    - E.g., CPU, DB, Disk, processes, sockets
  - Large botnets are available

Threat Model

- In scope
  - Attacks on higher layer bottlenecks, e.g., CPU, Memory, Database, Disk, processes, …
  - Attacks that fool the server to congest its uplink bandwidth
  - Mutating attacks
- Outside the scope
  - Flooding server's downlink (prior work)
  - Live-lock in the device driver

Tentative Solutions

- Filter big resource consumers?
- Passwords?
- Computational puzzles?

→No big consumers; Commodity OS do not support fine-grained resource accounting

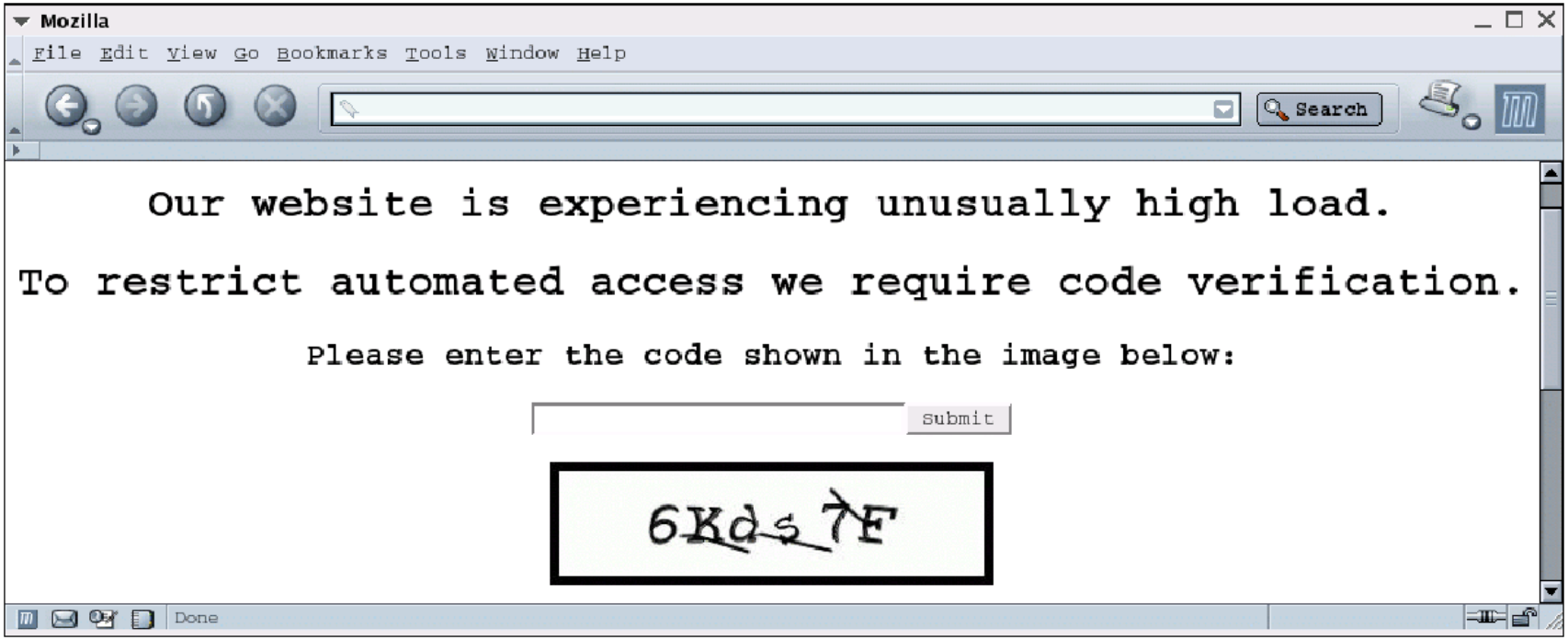
→Might not exist, expensive to check

→Computation is abundant in a botnet

????

Partial Solution:

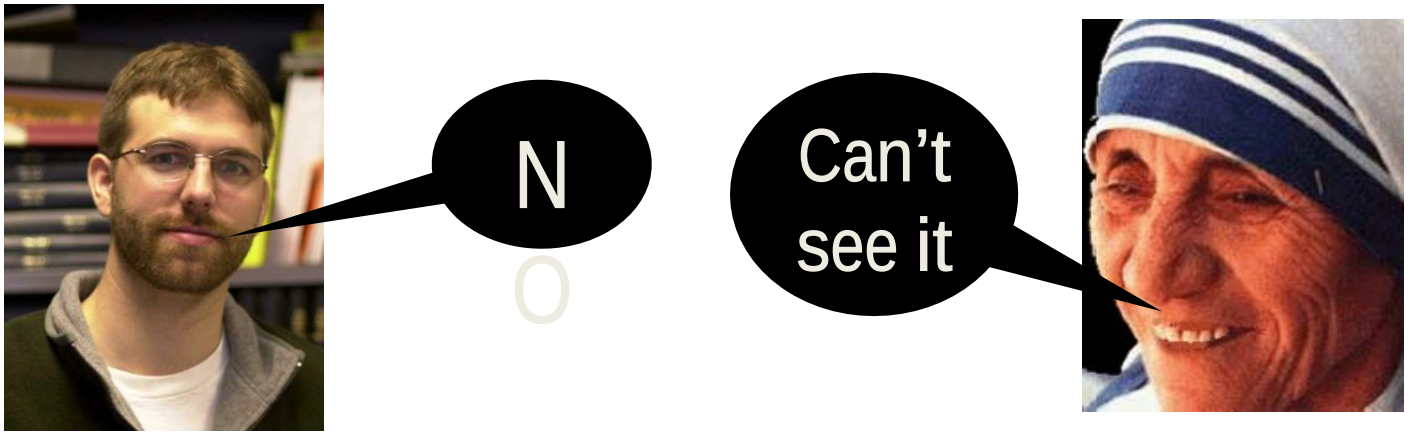
Reverse Turing Test (e.g., CAPTCHAs) to distinguish humans from zombies



But…

3 Problems with CAPTCHA Authentication

- (1) DDoS the authentication mechanism (connect to server, force context-switches, hog sockets etc.)
- (2) Bias against users who can't or won't answer CAPTCHAs

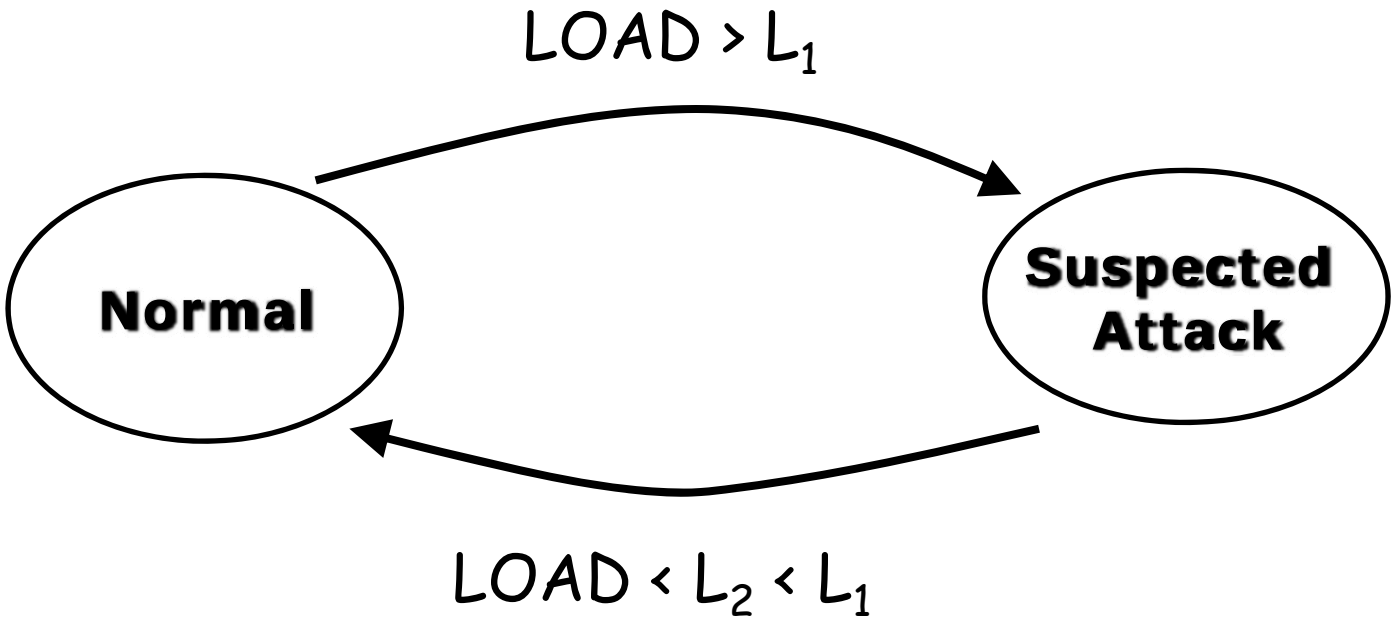


- (3) How to divide resources between service and authentication as to maximize system goodput?

Kill-Bots' Contributions

- First to protect against CyberSlam
- Solves problems with CAPTCHAs:
  - Cheap stateless authentication
  - Serves legit. users who don't answer CAPTCHAs
  - Optimal balance between authentication & service
- Improves performance during Flash Crowds
- Order of magnitude improvement in goodput & response time

Kill-Bots is a kernel extension for web servers

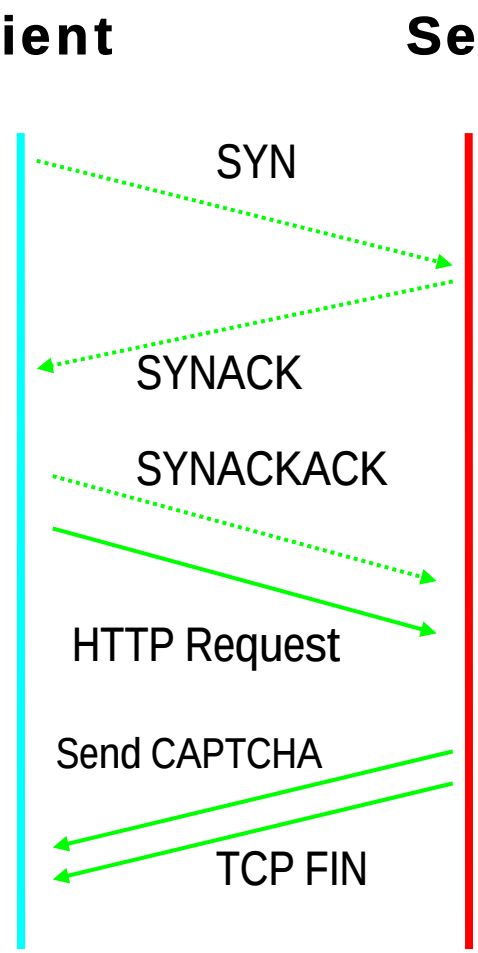


No Overhead

New Clients are authenticated once and given HTTP Cookie

Problem 1: Authentication vulnerable to DDoS

Problem 1: Authentication vulnerable to DDoS



Standard Network Stack

SYN Cookie

Check cookie, socket, reserve buffers
Causes context switch, buffer copies

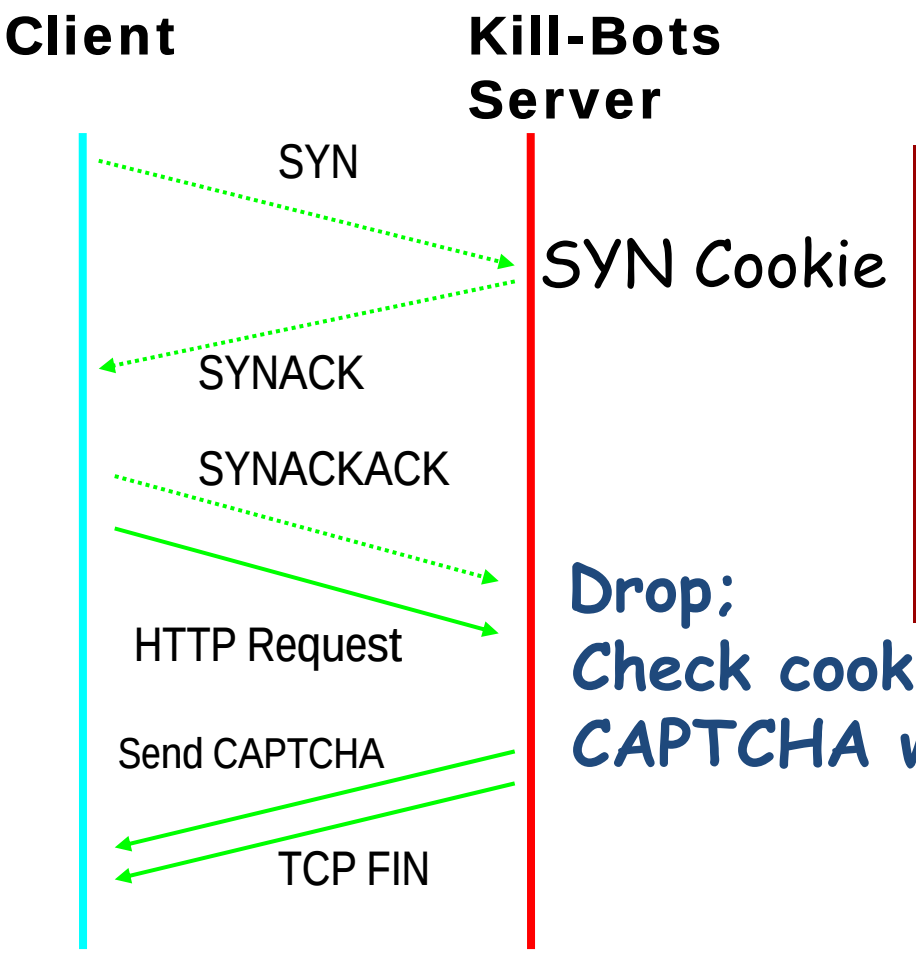
Resources are reserved till client sends a FIN but zombies don't FIN

Problem 1: Authentication vulnerable to DDoS

Solution: Modify network stack to issue CAPTCHAs without state

Problem 1: Authentication vulnerable to DDoS

Solution: Modify network stack to issue CAPTCHAs without state



Modified Network Stack

- Stateless & Cheap
- Keep congestion control semantics
- No browser mods.

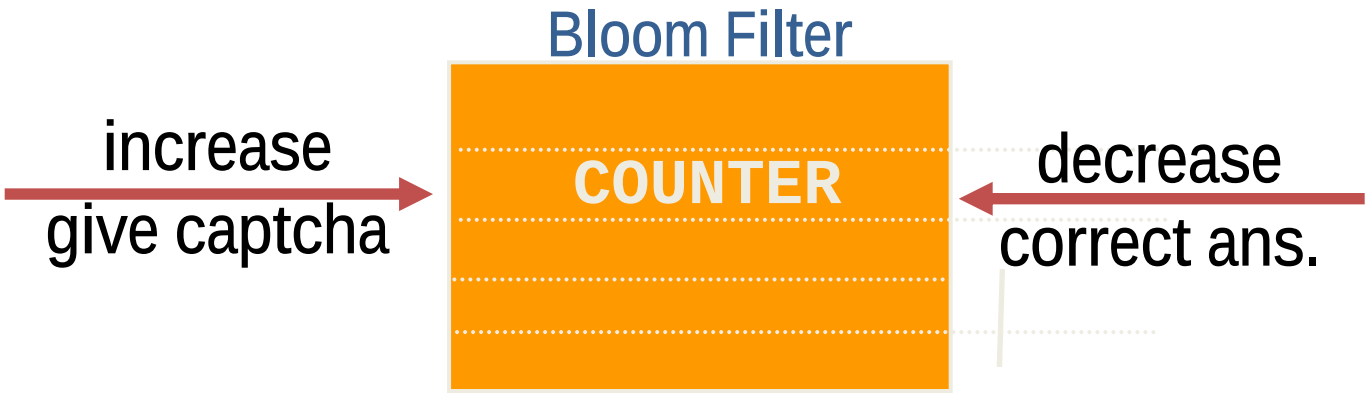
Drop;
Check cookie, send CAPTCHA without a socket!

Problem 2: Legit. Users who don't answer CAPTCHA

Solution: Use reaction to CAPTCHA

Humans	Zombies
(1) Answer CAPTCHA (2) Reload; if doesn't work, give up	Can't answer CAPTCHA, but have to bombard the server with requests

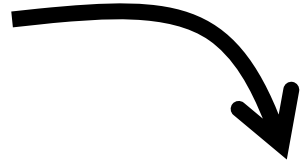
- Count the unanswered CAPTCHAs per IP, and drop if more than T; Cheap with a Bloom Filter



Stage 1:

- **CAPTCHA Authentication**
- Learn IP addresses of zombies using Bloom filter

Bloom Learns All Zombie IPs



Stage 2:

- Use only Bloom filter for Authentication
- No CAPTCHAs

Users who don't answer CAPTCHAs can access the server despite the attack in Stage 2

Problem 3: To Authenticate or To Serve?

Problem 3: To Authenticate or To Serve?

- Authenticate all new arrivals

  → can't serve all authenticated clients

- Authenticate very few arrivals

  → too few legitimate users are authenticated

Solution:

- Authenticate new clients with prob. α (drop others)

  → A form of admission control with 2 arrival types

But what α maximizes goodput?

Analysis

Modeled system using Queuing Theory

Found Optimal α^* (proof in paper)

But α^* depends on many unknown parameters

- attack rate
- mean service time
- mean session size
- legitimate request rate, etc…

Solution to Problem 3:

Kill-Bots adapts the authentication prob. by measuring fraction of time CPU is idle

Solution to Problem 3:

Kill-Bots adapts the authentication prob. by measuring fraction of time CPU is idle

- Analysis says: if idle > 0, α is prop. to (1- idle)
- Say you want to keep server busy 90% of time:

$$\frac{\alpha_{90\%}}{\alpha_{current}} = \frac{0.9}{1 - idle_{current}}$$

Solution to Problem 3:

Kill-Bots adapts the authentication prob. by measuring fraction of time CPU is idle

- Analysis says: if idle > 0, α is prop. to (1- idle)
- Say you want to keep server busy 90% of time:

$$\frac{\alpha_{90\%}}{\alpha_{current}} = \frac{0.9}{1 - idle_{current}}$$

- Kill-Bots adapts in real time

$$\alpha_{90\%} - \alpha_{current} = \alpha_{current}\left(\frac{idle_{current} - 0.1}{1 - idle_{current}}\right)$$

Solution to Problem 3:

Kill-Bots adapts the authentication prob. by measuring fraction of time CPU is idle

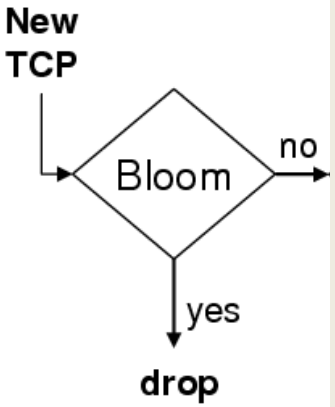
- Analysis says: if idle > 0, α is prop. to (1- idle)
- Say you want to keep server busy 90% of time:

$$\frac{\alpha_{90\%}}{\alpha_{current}} = \frac{0.9}{1 - idle_{current}}$$

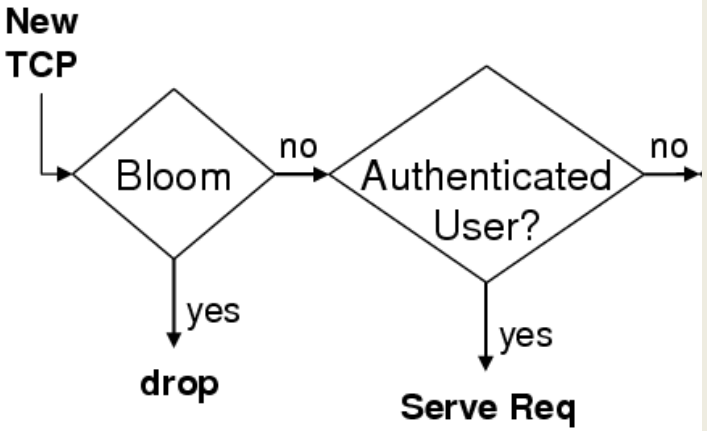
- Kill-Bots adapts in real time

$$\Delta\alpha = \frac{1}{8}\alpha_{current}\left(\frac{idle_{current} - 0.1}{1 - idle_{current}}\right)$$

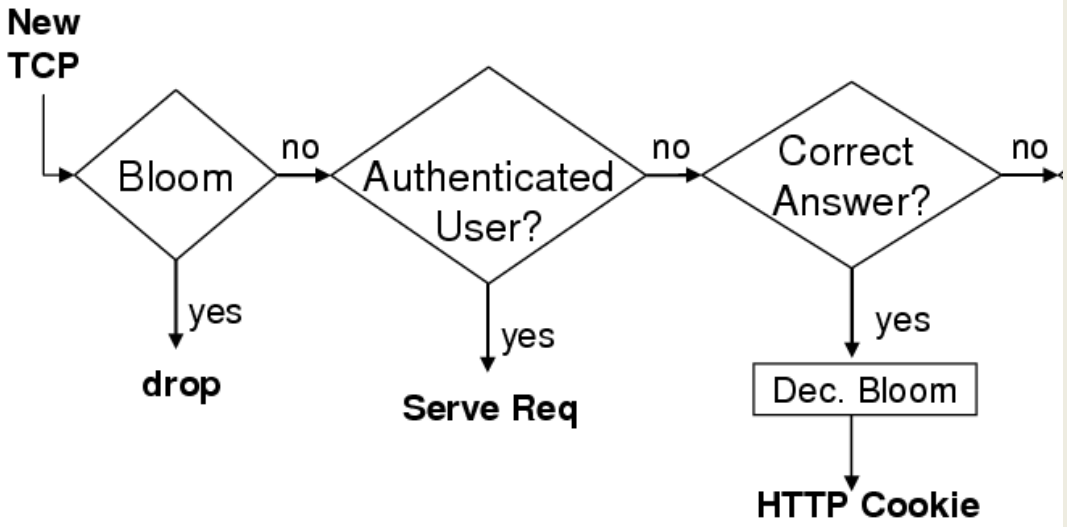
Tying it Together



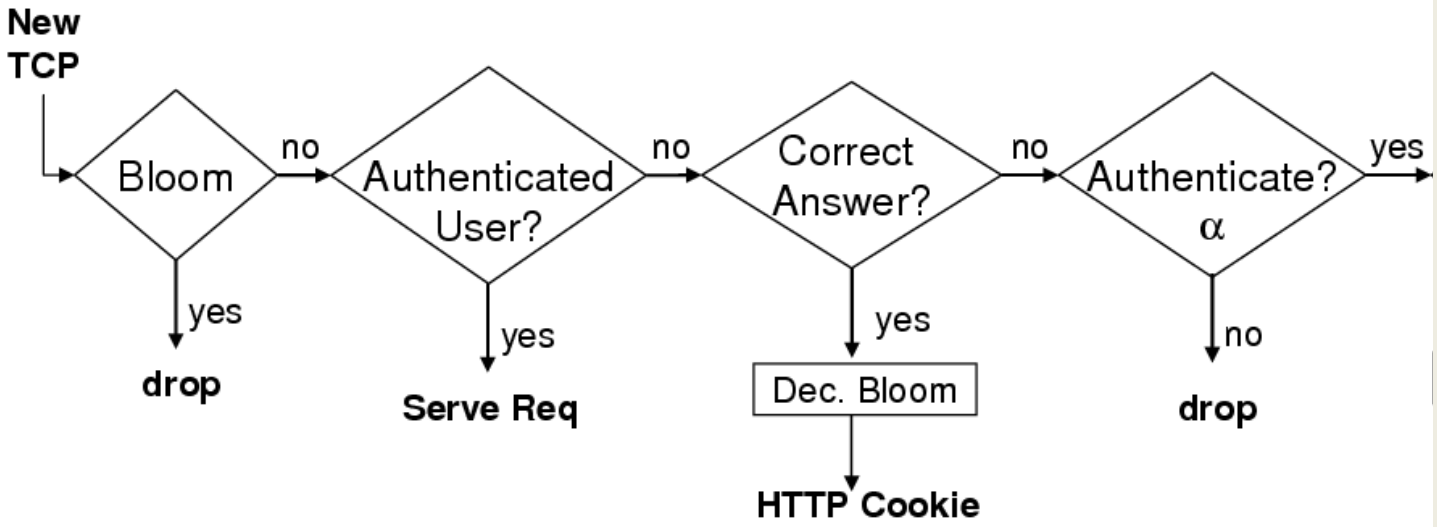
Tying it Together



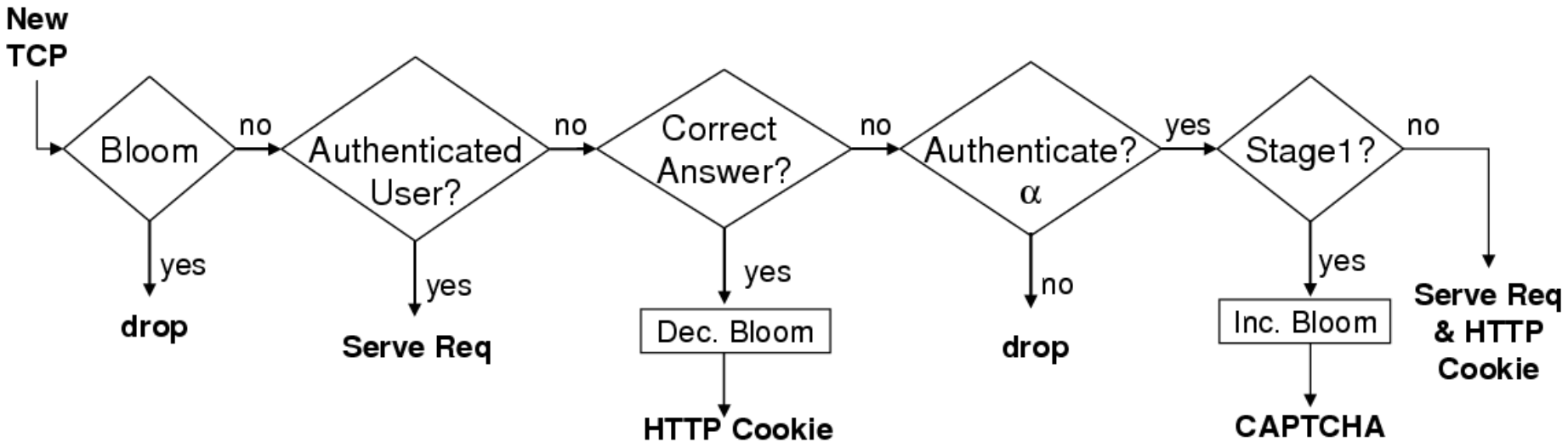
Tying it Together



Tying it Together



Tying it Together



Recap: Kill-Bots addresses CyberSlam

Problem

- DDoS the authentication
- Serve legitimate users who don't answer CAPTCHAs
- Divide resources between authentication & service

Solution

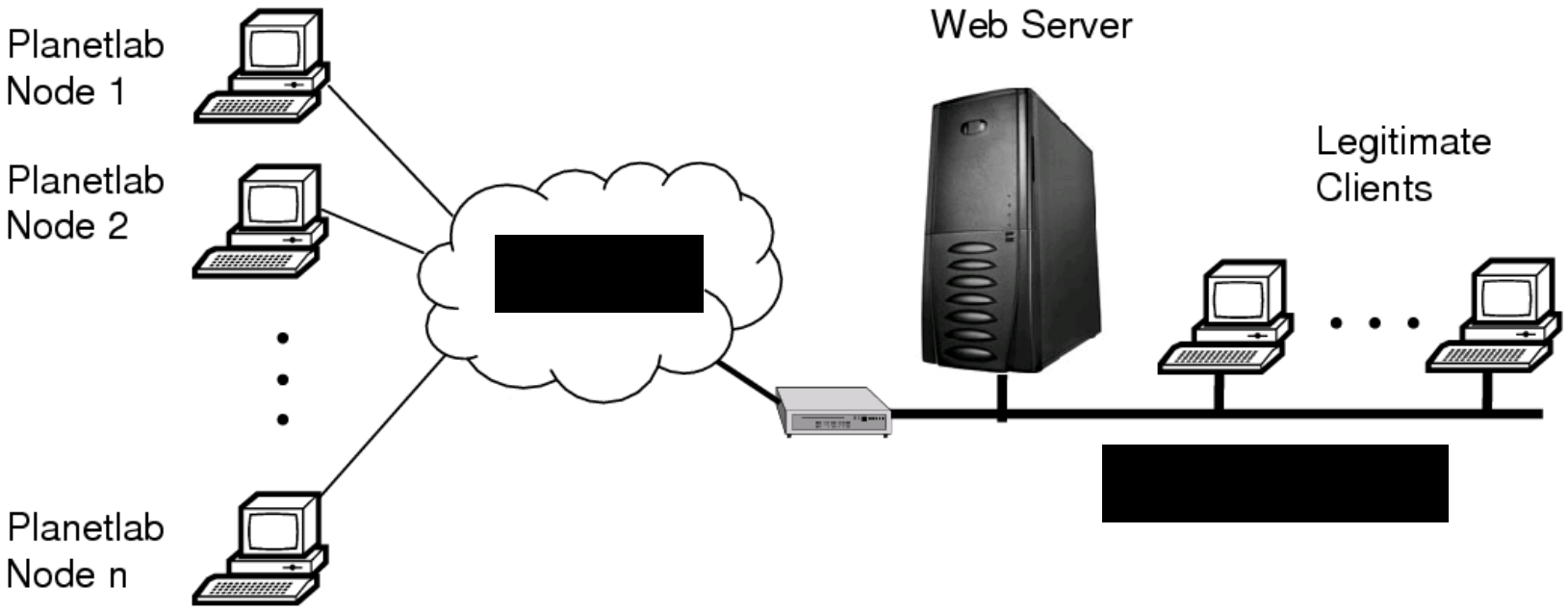
- Send CAPTCHAs cheaply without sockets
- Use reaction to CAPTCHA to identify zombies
- Adaptive authentication as admission control

Attacks & Defenses

- Replay Attacks?
  - Don’t work. Limit #connections per cookie
- Spoof IP, cause Bloom filter to block
  - Doesn’t happen. SYN cookie before updating Bloom
- Breaking the CAPTCHA?
  - Kill-bots can use any Reverse Turing Test

Performance

Wide-area Evaluation Using PlanetLab

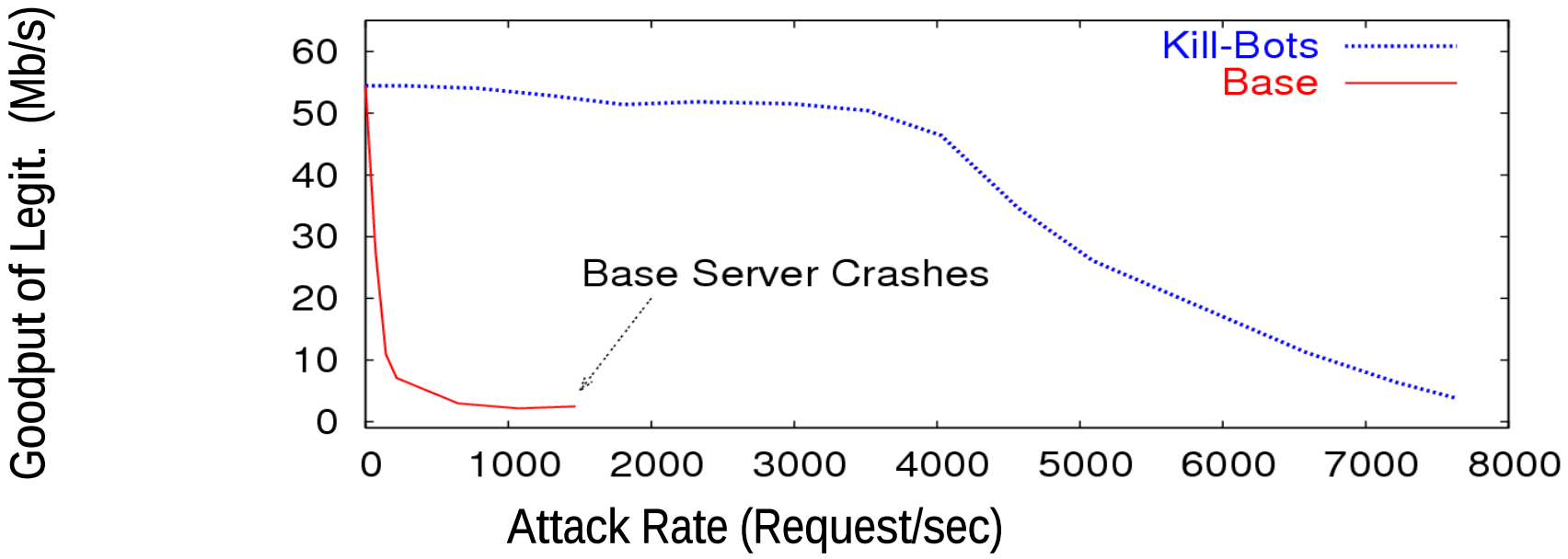


- Legit. users are driven from CSAIL Web traces
- >25,000 attackers on PlanetLab request random pages
- 60% of legitimate users answer CAPTCHAs

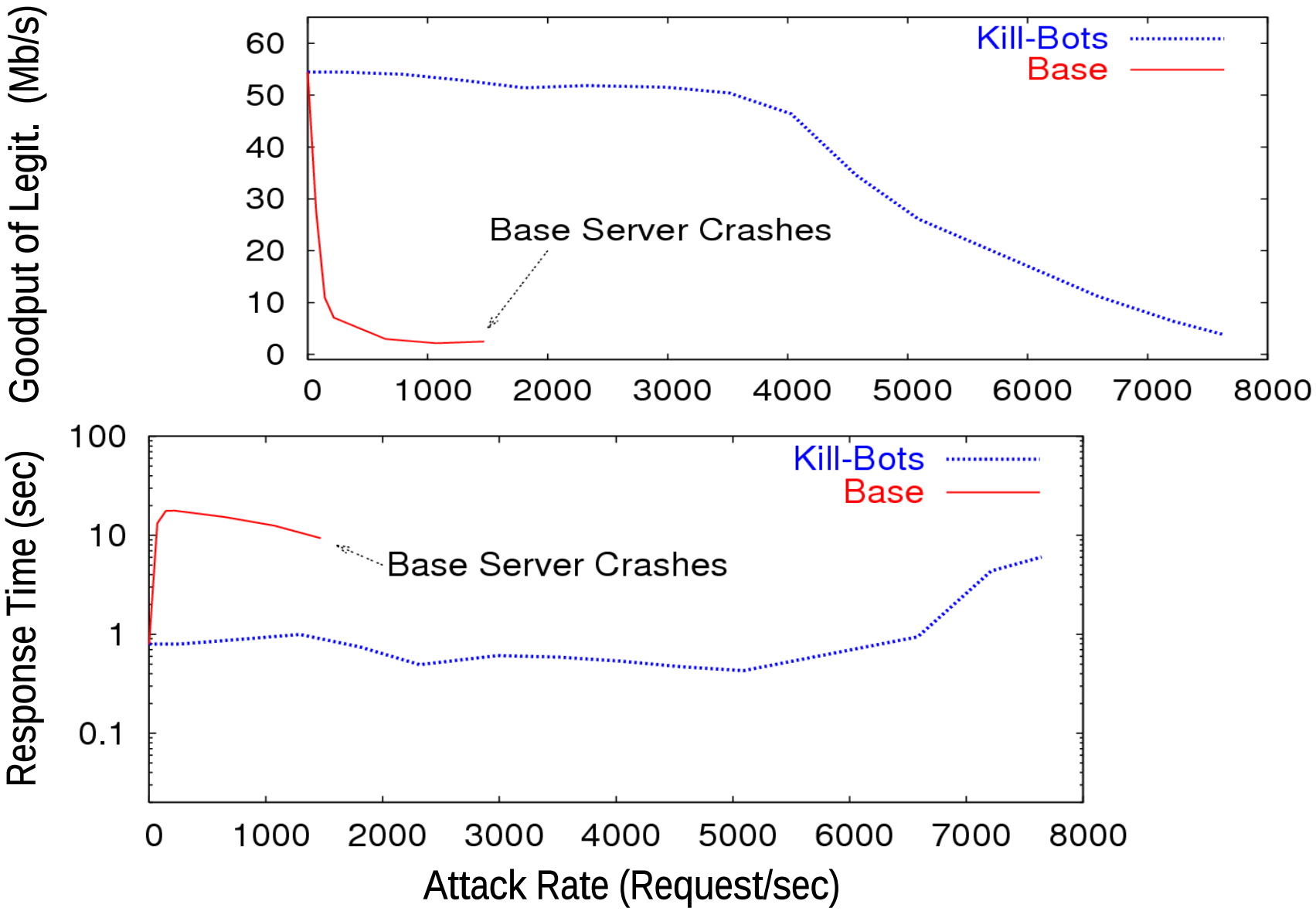
Metrics

- Goodput (of Legitimate Users)
- Response time (of Legitimate Users)
- Maximum survivable attack rate

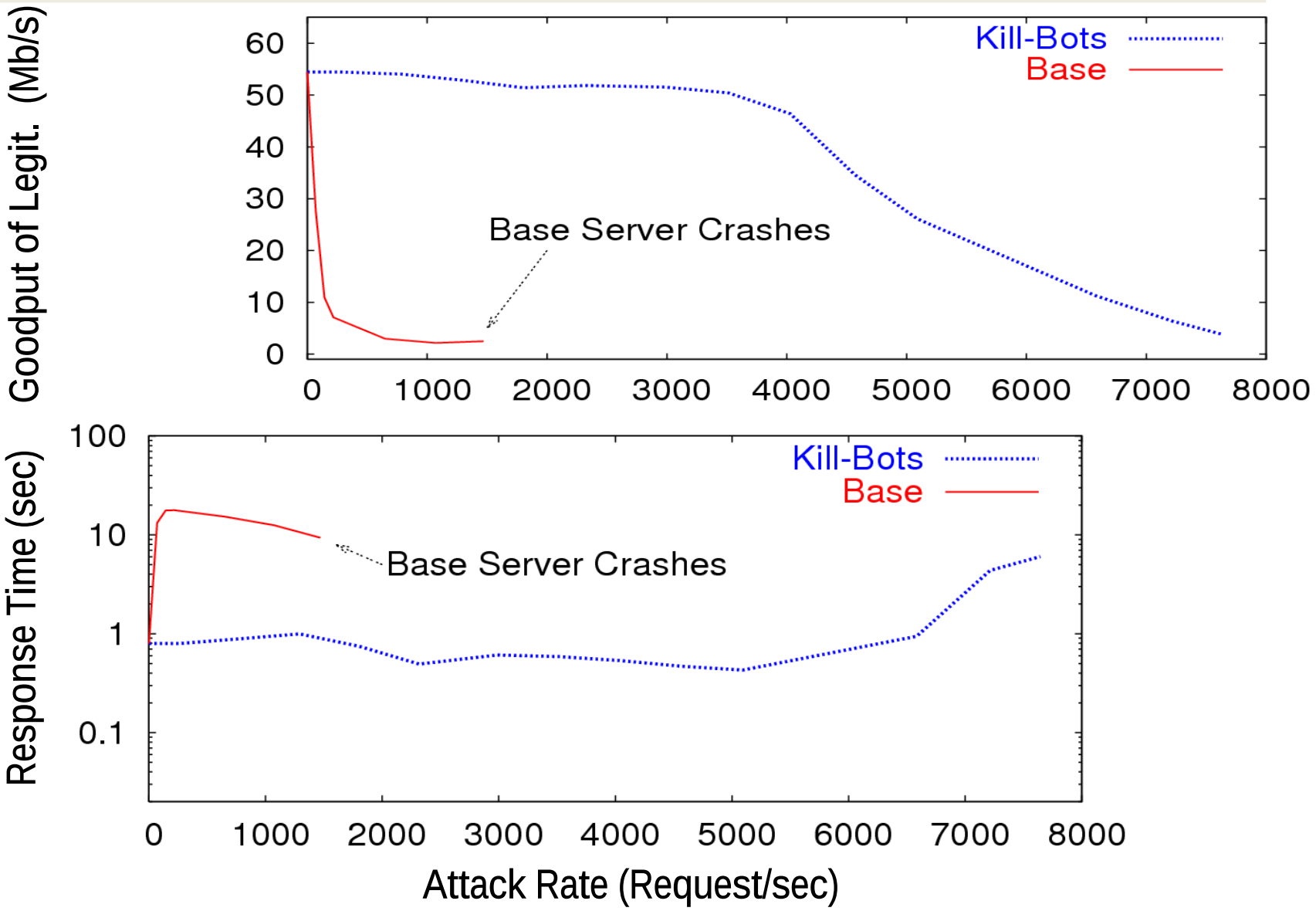
Kill-Bots under DDoS



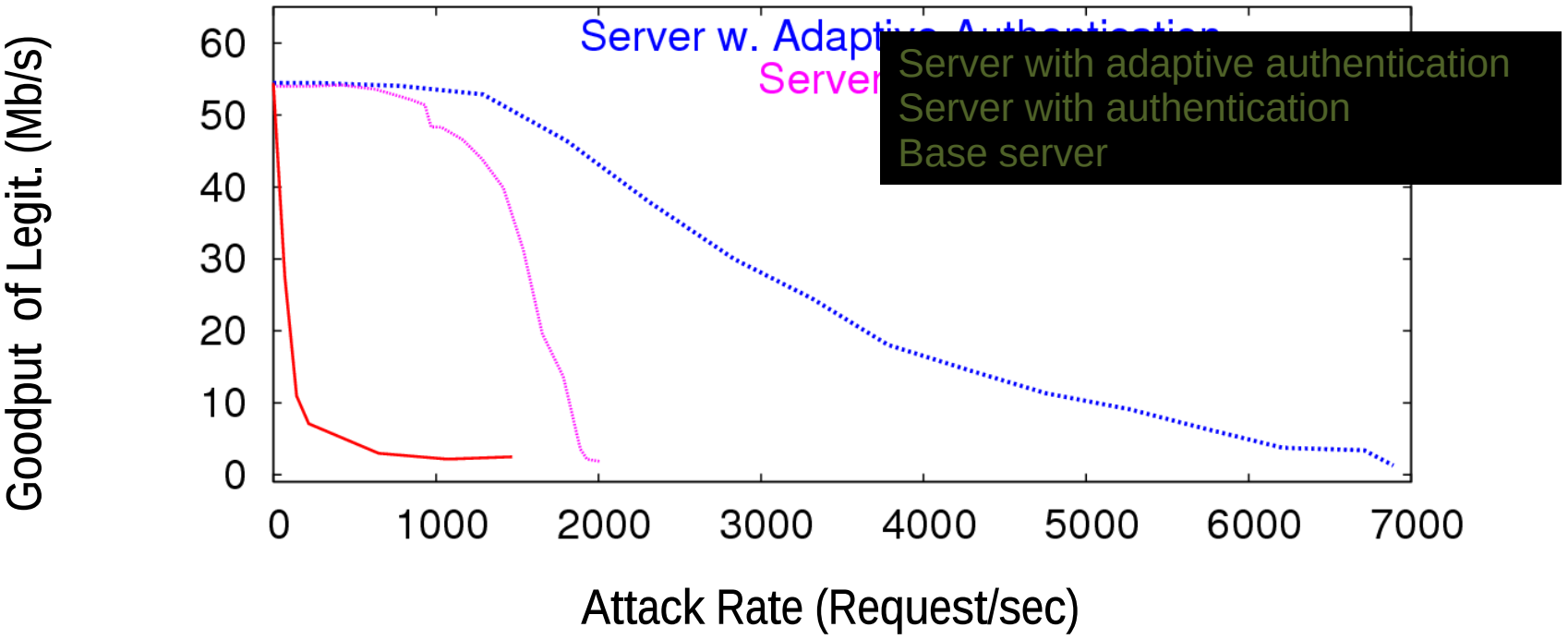
Kill-Bots under DDoS



5-10 times better Goodput and Response Time

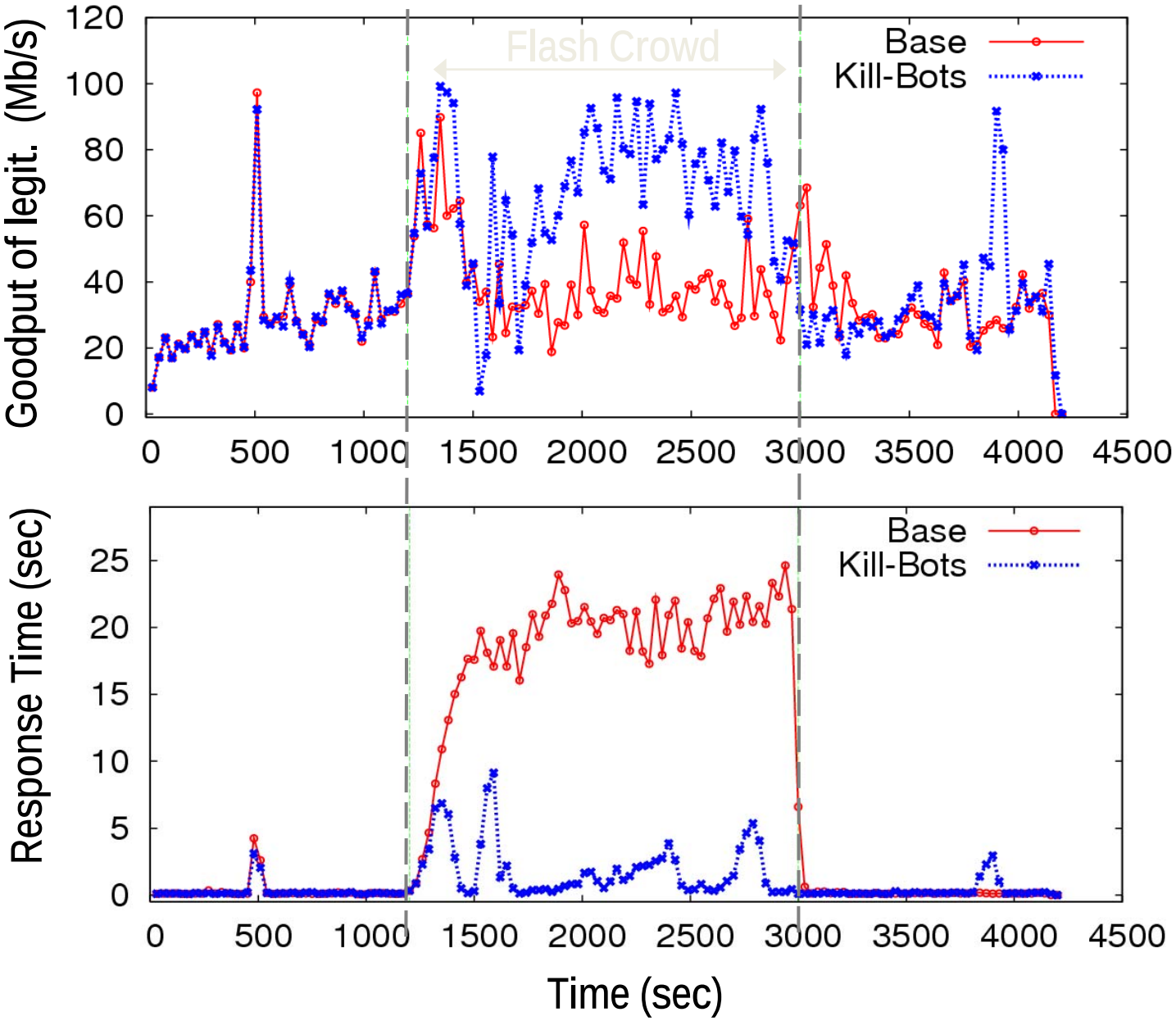


Why Adapt the Authentication Probability?

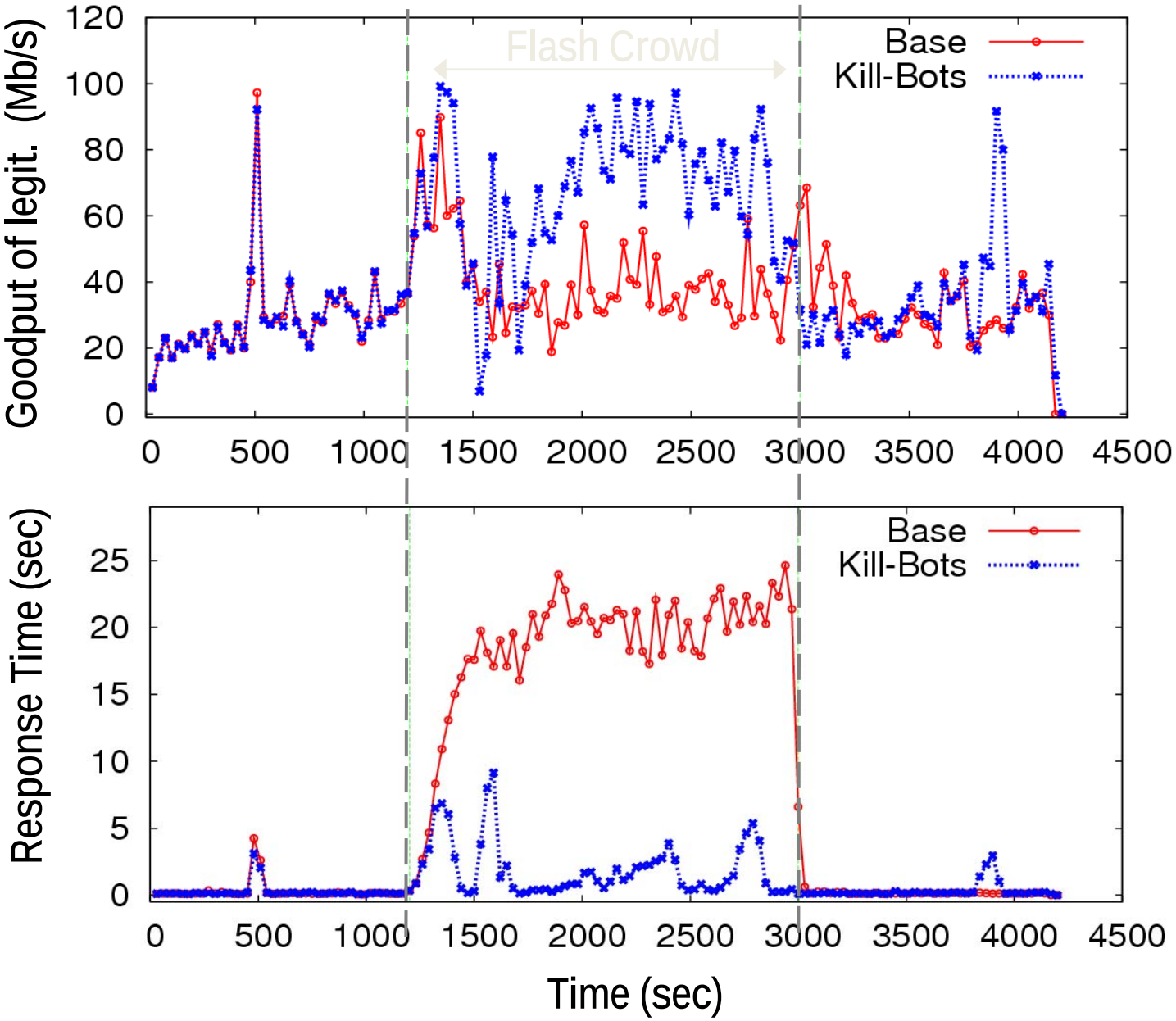


Adaptive α is much better than authenticating every new user

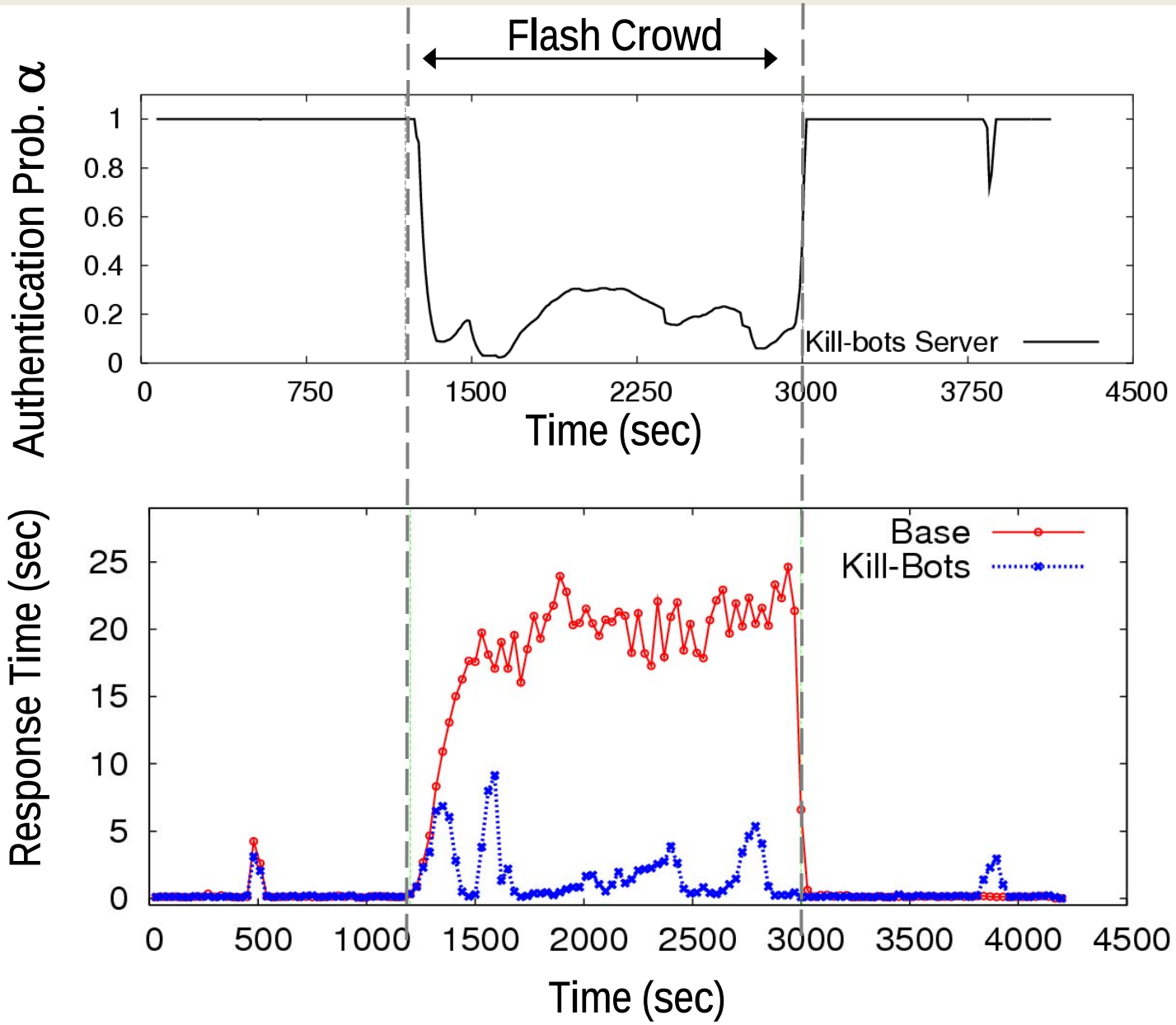
Kill-Bots under Flash Crowd



Orders of magnitude better Response Time

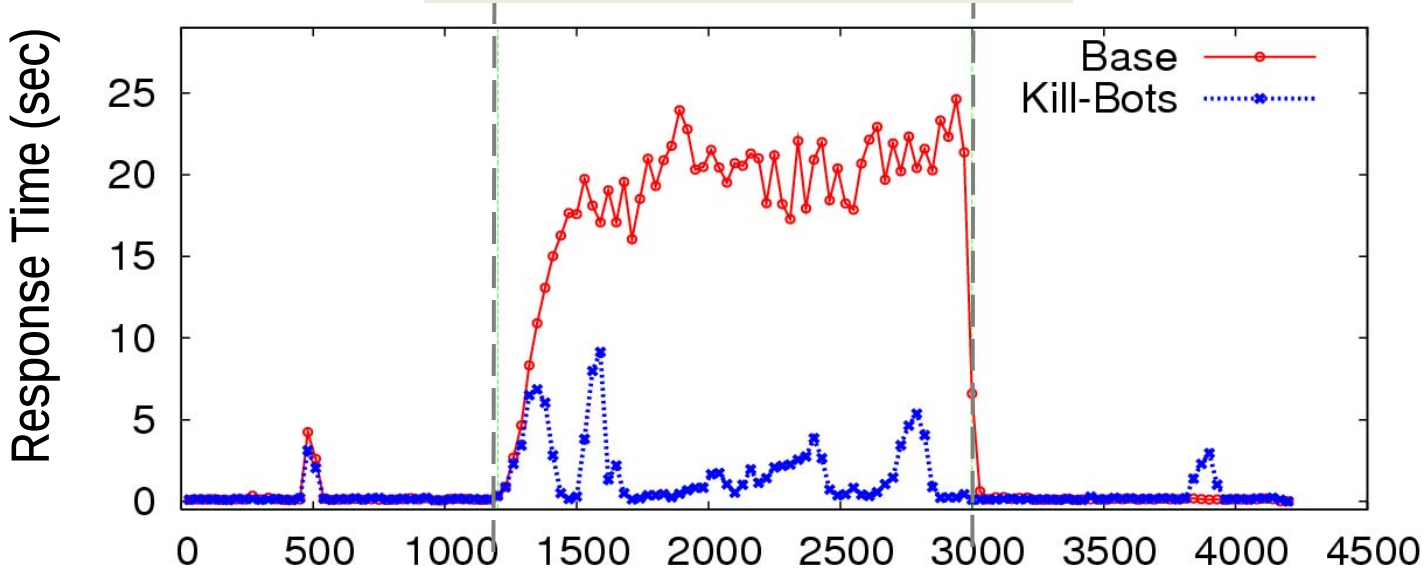


Adaptive α provides admission control



Kill-Bots under Flash Crowd

	Base Server	Kill-Bots
Number of dropped legitimate requests	360,000	80,000



Kill-Bots authenticates new clients only if it can

Kill-Bots' Contributions

- First to protect Web servers from DDoS attacks that mimic legitimate browsing
- First to deal with CAPTCHA's bias against legitimates users who don't solve them
- Sends CAPTCHA and checks answer without any server state
- Addresses both DDoS attacks and Flash Crowds
- Orders of magnitude better response time, goodput, and survivable attack rate