CDN on Demand

Affordable DDoS Defense using Untrusted IaaS-Clouds

Yossi Gilad, Michael Goberman, Amir Herzberg and Michael Sudkovitch
Talk Outline

- Content Delivery Networks as DoS defense
- The CDN-on-Demand system
  - Clientless secure objects
  - Loss resilient tunnel
- Performance evaluation
CDN as a DoS Defense
CDN as a DoS Defense

Many clients → Content-Origin
CDN as a DoS Defense

- Host site on Content Delivery Network (CDN)
  - Distribute content from multiple, geo-dispersed proxies
  - High-bandwidth, distributed and scalable infrastructure
- But there are problems…

Diagram:
- Many clients
- Proxy 1
- Proxy 2
- Proxy 3
- Content-Origin
- Content
- Content
- Content
CDNs against DoS: Problems

- **Cost**
  - CDNs provide `continuous, full service’ ➔ expensive
  - Service sometimes unavailable to small sites
- **Disclose keys (HTTPS sites)**
  - Threat model: CDN servers may be malicious/compromised
- **Tradeoff: Cheaper CDNs may be less secure/trusted**
  - Akamai/Amazon vs. CDN77 ➔ 10X difference in cost

**Can we build a secure & low-cost CDN-based defense?**
CDN-on-Demand: Overview

- A CDN system built on multiple low-cost IaaS clouds
  - Deploys proxies only when/where needed
- Object level security, avoid sharing keys with CDN
- Software package, rather than third-party service
  - Open source www.autocdn.org
  - Anyone can install
CDN-on-Demand: Overview

- Cloud 1
  - Clients
  - Content-Origin
  - Gateway
  - Monitor
  - Watchdog

- Cloud 2
CDN-on-Demand: Overview

Many clients → Cloud 1 → Content-Origin

Cloud 1

Content-Origin

Cloud 2
CDN-on-Demand: Overview

Many clients → Cloud 1 → Content-Origin

Cloud 1
- proxy 1
- watchdog

Cloud 2
- proxy 2
Security: Why not just use TLS?

Many clients → Cloud 1 → TLS → proxy 1 → Content-Origin

Cloud 2 → TLS → proxy 2 → Cloud 1 → content

TLS: Transport Layer Security

Cloud: Cloud Computing Infrastructure

Proxy: Network Device used to increase the security of a network

Content-Origin: The Source of the Content
Clientless Secure Objects

- Idea: store `secure objects’ on untrusted proxies
  - Don’t share private keys
  - Complement TLS network level protection
  - Restriction: avoid changes to clients
- Important flexibility for `on-demand’ system
  - Allows to use cheaper, less trusted clouds
  - Allows to switch between clouds
Setup (once per month)

- Client
  - Get / (TLS connection)
  - Loader script
    - Get root.js
    - \( H(d) = d \)
    - stay in cache

- CDN proxy
  - site.cdn.com

- Gateway
  - homepage

- Content-Origin
  - site.com
Content Distribution

Client

Get homepage

CDN proxy

Get embedded object

Gateway

Content-Origin

Verify and present

Verify and present

PK

PK

Content-origin not involved
Clientless Secure Objects: Computation

- JavaScript crypto is inefficient
  - Over 20X time for signature verification cf. native code (RSA2048)
  - Single threaded computations
  - Significantly delays content display time
- Observation: most of the time loading an object is spent waiting for its data to arrive
- Compute incrementally utilizing Merkle-Damgard

\[
\sigma \quad d \quad \text{data 1} \quad \text{data 2} \quad \text{data 2}
\]

\[h = d?\]
Clientless Secure Objects: Performance

- Tested using content from popular homepages
- 2% overhead for page load-time
  - Incremental processing reduces overhead approx. 70%
Delivering Content Updates under DoS

Many clients → Cloud 1

Cloud 1

proxy 1

watchdog

Cloud 2

proxy 2

Content-Origin

content
Loss-Resilient Tunnel

- Tunnel packets between content-origin (via gateway) and proxies over UDP
  - Client connects via HTTP(S) -- no changes to clients
- Use network coding to ensure delivery even with high loss, e.g., [Rabin 89’]
  - Recover from loss if n-out-of-m packets arrive

![Diagram of content-origin, client, proxy, and loss-resilient tunnel]
Loss-Resilient Tunnel

![Graph showing response time against attack rate/link capacity](image)

- 1. Response Time
- 2. Response Time, Loss-Resilient Tunnel

**Graph Details:**
- **Y-axis:** Time in seconds (log-scale)
- **X-axis:** Attack Rate / Link Capacity
- The graph demonstrates the relationship between response time and attack rate/link capacity, highlighting the resilience of the tunnel under varying conditions.
Evaluation

- Deployment over EC2 and GCE
- PlanetLab clients download 50KB object repeatedly
- Monitor performance while introducing changes to the setting every few minutes
  - more clients, server crash, attack on origin…
Results

• Handle thousands of clients simultaneously
• Attacks on content-origin have limited effect
  • due to loss-resilient tunnel
• Fraction of the cost of commercial CDN defenses
Questions?

Thank you 😊