Distributed Denial of Service (DDoS): Attacks and Defenses

I. Pustogarov

Forwarding-Loop Attacks in Content Delivery Networks

NDSS 2016

http://www.icir.org/vern/papers/cdn-loops.NDSS16.pdf

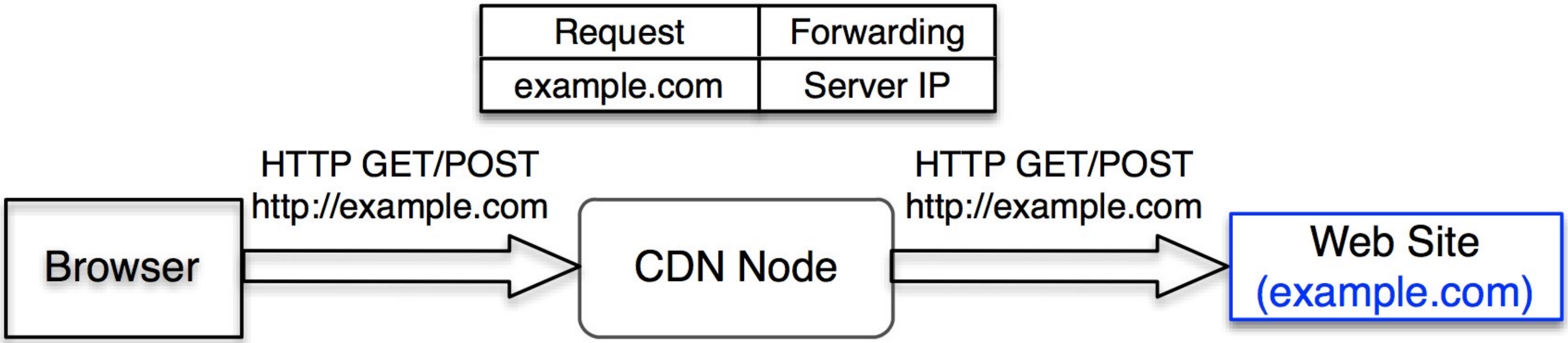
Summary

- Present "forwarding loop" attacks that threaten CDN availability
- A case that highlights the danger of allowing cross-organization, user-controlled (untrusted) policies without centralized administration
- Measured 16 popular CDNs and find all of them are vulnerable to such attacks
- Vendors have acknowledged the problem

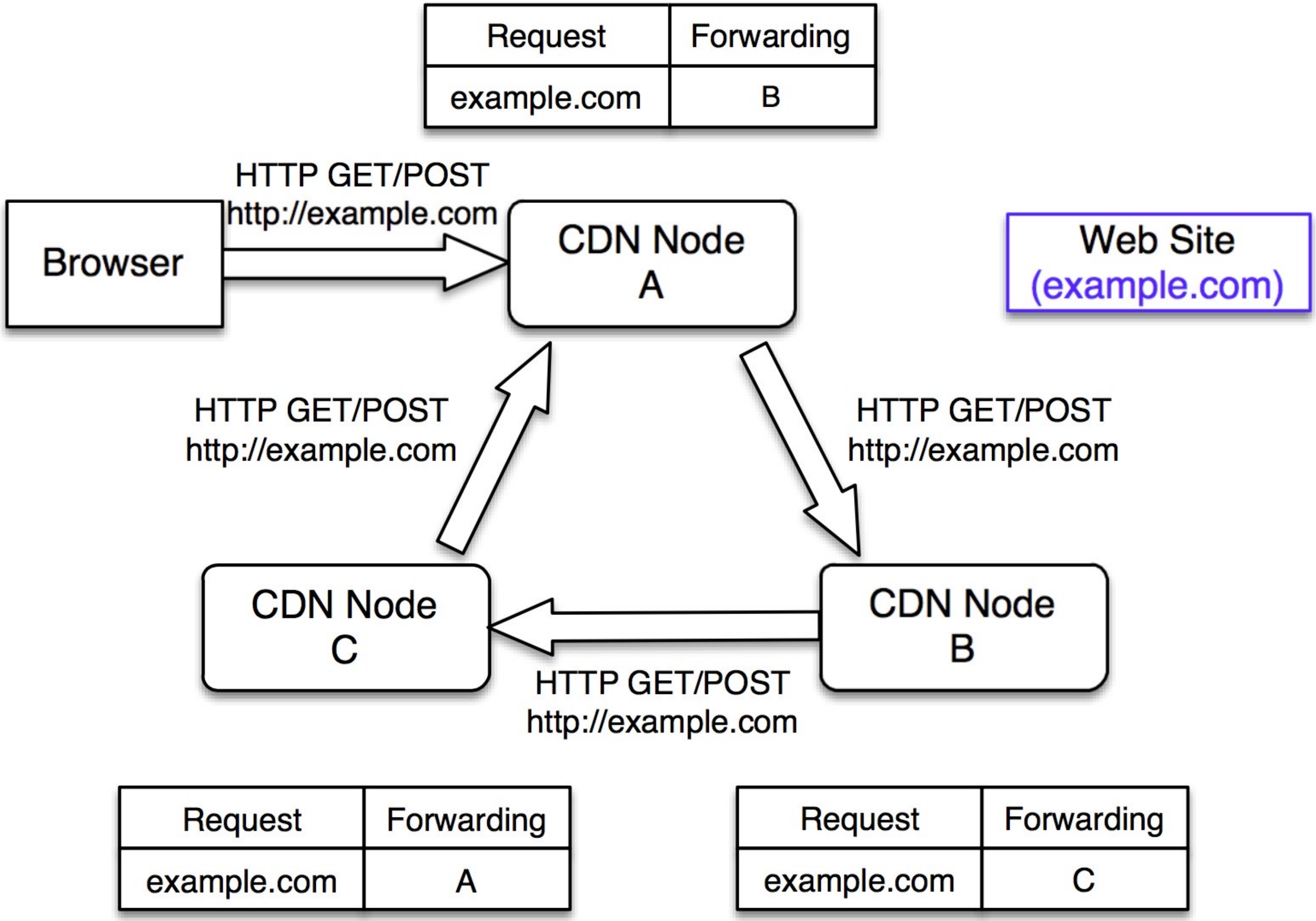
Tested CDNs

Baidu 百度
CLOUDFLARE
fastly
edgecast
Tencent
Alibaba.com
CDN77.com
CDN sun
anquanbao.com
Akamai
Azure (China)
amazon cloudfront

Normal CDN forwarding rule



CDN Forward loop creation



Attack practicality

- Cost : All 16 CDNs provide free or free-trial account
- Can remain anonymous: 11/16 CDNs only require an email address
- Some CDNs agreed this attack is severe

Defenses

- Unifying and standardizing a loop-detection header (“Via” as recommended by RFC)

- Interim defenses
  - Obfuscating self-defined loop-detection headers
  - Monitoring and rate-limiting
  - Constraint on forwarding destination

Mirai botnet and DDoS attacks

Understanding the Mirai Botnet

(USENIX Security 2017)

https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf

- Starting in September 2016, several massive DDoS attacks were launched from the Mirai botnet

- Victims: Krebs on Security (620 Gbps), OVH (1Tbps), and Dyn (affected: GitHub, Twitter, Reddit, Netflix, Airbnb)

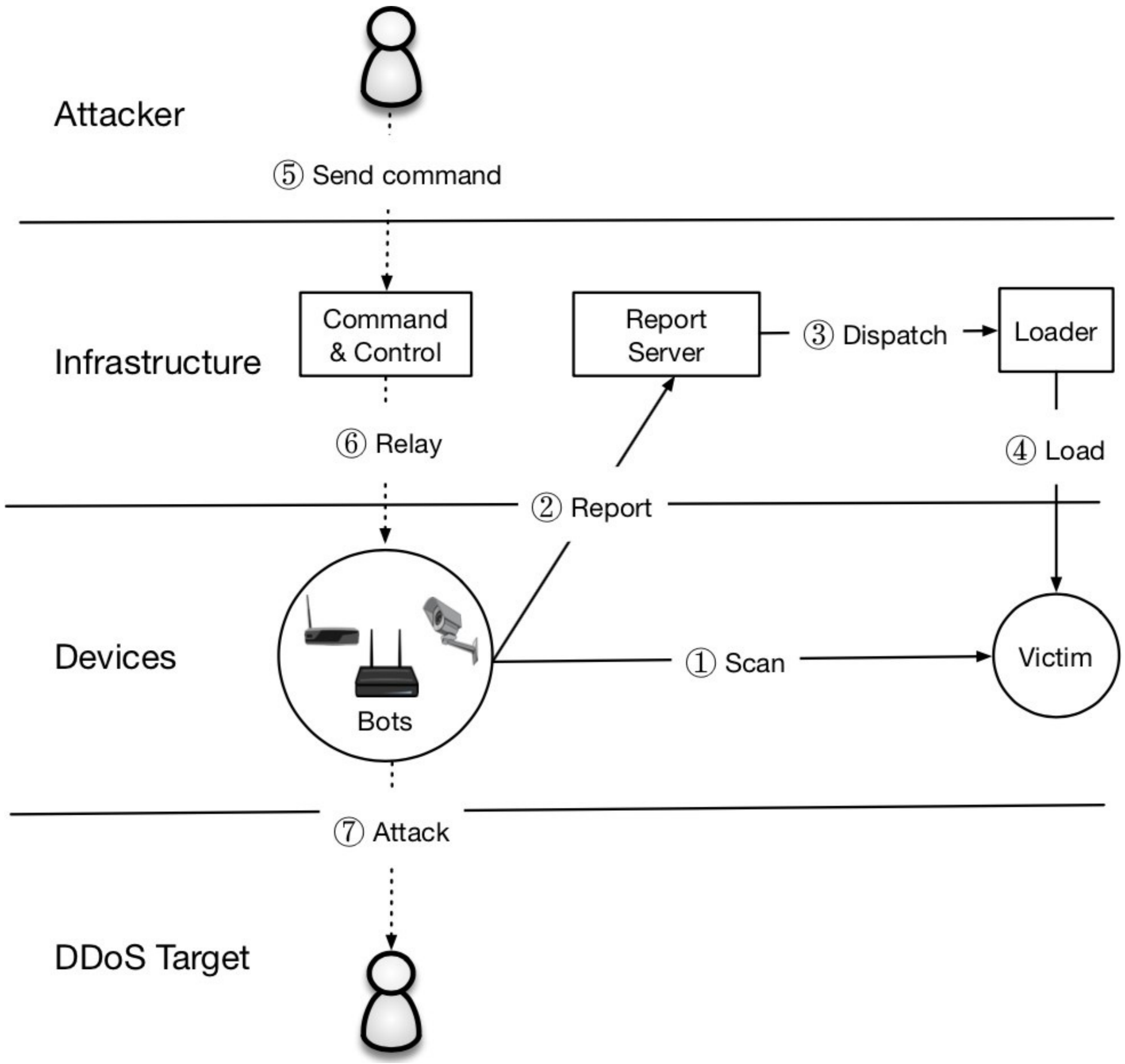
- Abused hundreds of thousands of some of the Internet's least powerful hosts — IoT devices

- The Mirai botnet, composed primarily of embedded and IoT devices: DVRs, IP cameras, routers, and printers

- Infected nearly 65,000 IoT devices in its first 20 hours before reaching a steady state population of 200,000–300,000 infections; peak infection: 600,000

- Features: efficient spreading based on Internet-wide scanning, rampant use of insecure default passwords in IoT products, and the insight that keeping the botnet's behavior simple would allow it to infect many heterogeneous devices

Mirai Operation



Mirai Operation

- Starts with: *the rapid scanning* phase "statelessly" sent TCP SYN probes to pseudorandom IPv4 addresses, excluding those in a hard-coded IP blacklist, on Telnet TCP ports 23 and 2323

- If a potential victim is found, starts a *brute-force login* phase: a Telnet connection using 10 username and password pairs selected randomly from a pre-configured list of 62 credentials.

- At the first successful login, Mirai sent the victim IP and associated credentials to a hard-coded *report server.*

- A separate *loader program* asynchronously infected these vulnerable devices by logging in, determining the underlying system environment, and finally, downloading and executing architecture-specific malware.

Mirai Operation

- After a successful infection, Mirai attempted to conceal its presence by deleting the downloaded binary and obfuscating its process name in a pseudorandom alphanumeric string.
- As a consequence, Mirai infections did not persist across system reboots.
- It also killed other processes bound to TCP/22 or TCP/23, as well as processes associated with competing infections, including other Mirai variants

DDoS attack types

Attack Type	Attacks	Targets	Class
HTTP flood	2,736	1,035	A
UDP-PLAIN flood	2,542	1,278	V
UDP flood	2,440	1,479	V
ACK flood	2,173	875	S
SYN flood	1,935	764	S
GRE-IP flood	994	587	A
ACK-STOMP flood	830	359	S
VSE flood	809	550	A
DNS flood	417	173	A
GRE-ETH flood	318	210	A

Table 9: **C2 Attack Commands**—Mirai launched 15,194 attacks between September 27, 2016–February 28, 2017. These include [A]pplication-layer attacks, [V]olumetric attacks, and TCP [S]tate exhaustion, all of which are equally prevalent.

DDoS attack types

- Only 2.8% of Mirai attack commands relied on bandwidth amplification

- Amplification attacks make up 74% of attacks issued by DDoS-for-hire booter services

Timeline

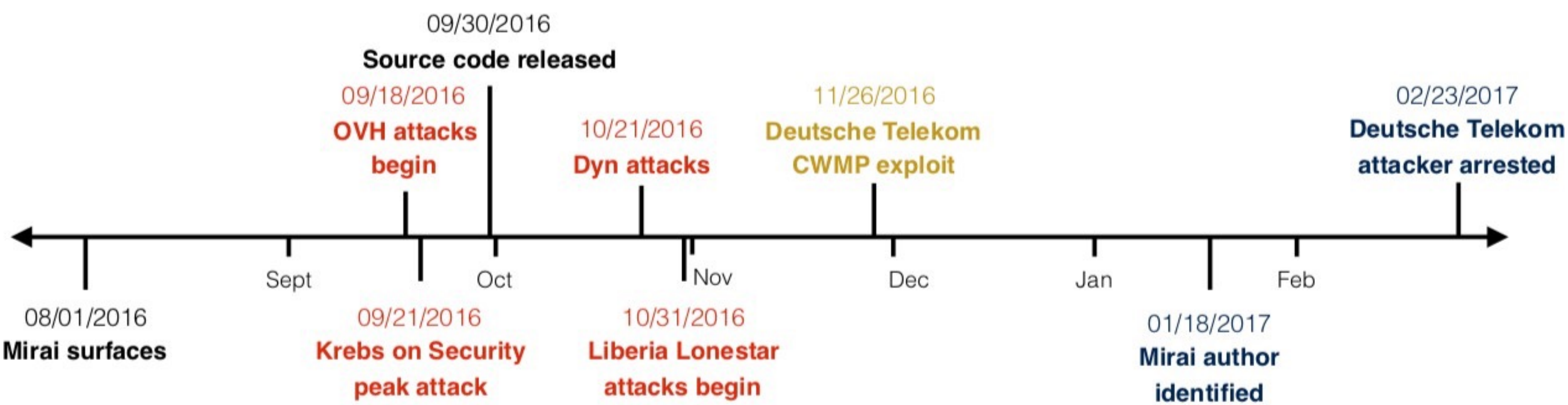


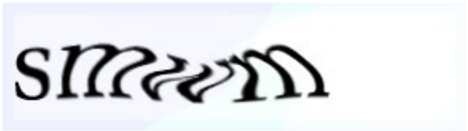
Figure 1: **Mirai Timeline**—Major attacks (red), exploits (yellow), and events (black) related to the Mirai botnet.

“Botz-4-Sale: Surviving Organized Attacks That Mimic Flash Crowds”, Kandula et al.

Kill-bots – a solution at the target

- Motivations
  - Bots can easily mimic flash crowds (differ in intent – but in content)
  - Requests originate from diverse geographic locations
  - Can authentication, puzzles help?

- Kill-bots
  - Kernel extension to protect web servers
  - Combine two functions:
    - Authentication
    - Admission control
  - Relies on: automated Turing tests (ATTs)
    - E.g.: CAPTCHAs



  - Also, relies on: attackers must launch a "fast" DDoS to succeed
    - Why so?

Kill-bots: Authentication

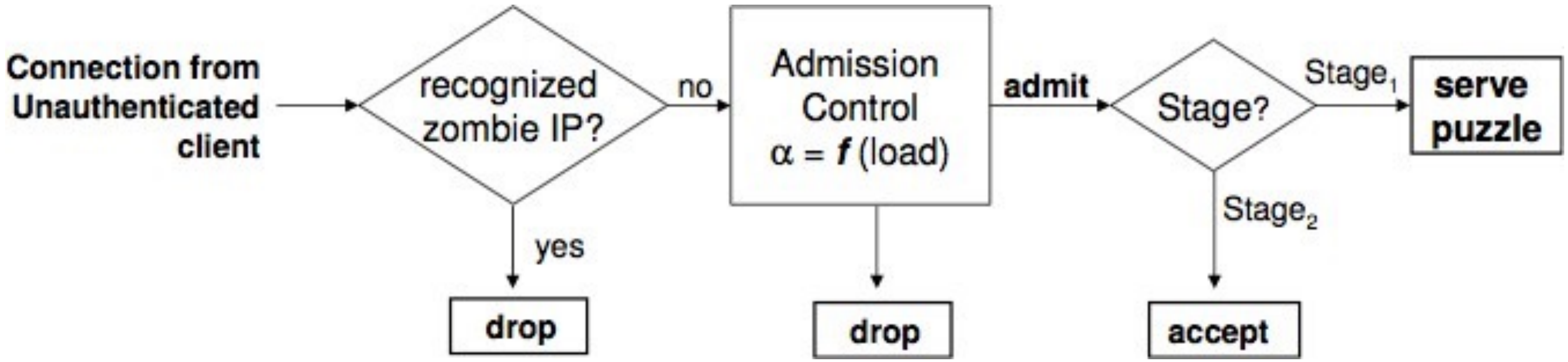
- Activated when the server is overloaded
- Two stages
- Stage 1:
  - Each new session must solve an ATT
  - Human users may solve ATTs – if they want service
  - Bots will be "unable" to solve ATTs
  - SYN-cookies are used to address spoofed IPs
  - Bloom filter is used to count how often an IP address failed to solve a test
  - The server discards requests from a client if the number of its unsolved tests exceeds a given threshold (e.g., 32)

Kill-bots: Authentication Stage 2

- Kill-bots switches to Stage2 after the set of detected zombie IP addresses stabilizes
  - The filter does not learn any new bad IP addresses
- ATTs are no longer served
- Kill-bots relies solely on the Bloom filter to drop requests from malicious clients (from stage 1)

- This allows legitimate users who cannot, or do not want to solve ATTs access to the server despite the ongoing attack

Kill-bots: an overview



Activating Kill-bot authentication

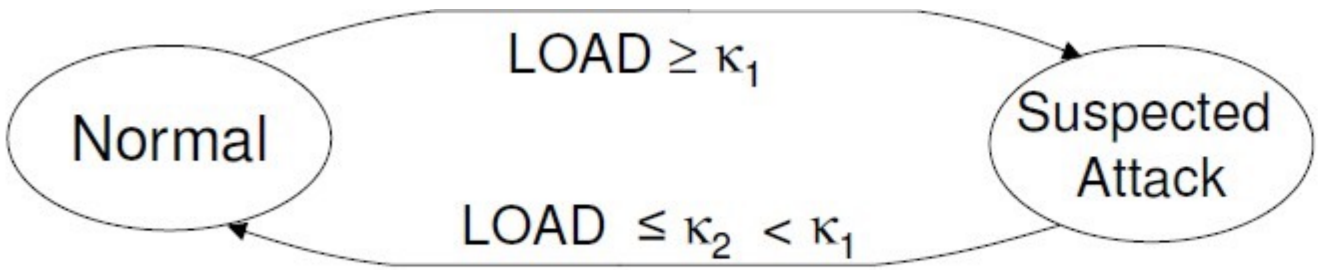


Figure 2: A Kill-Bots server transitions between NORMAL and SUSPECTED_ATTACK modes based on server load.

Security analysis of Kill-bots

1. Social engineering attacks
    - Use unsuspecting users to solve ATTs – effective against email / account creation ATTs
    - But in Kill-bots: ATT answers expire (after 4minutes)
        - i.e.: continuous authentication
2. Bloom-filter pollution: fails due to SYN cookies (no IP spoofing)
3. Copy attacks: limit in-progress requests to a low threshold (e.g., 8)
4. Database with answers to all ATTs
    - Add new ATTs to frustrate the attacker
5. Breaking ATTs
    - Design better ones