Arming the Defenseless: An Incentive-based Approach to DNS Reflection Prevention

Casey Deccio, Brigham Young University
AIMS 2017
CAIDA, UCSD, La Jolla, CA
March 1, 2017
Reflection/Amplication-based DDoS Attack

- Attackers (Globally distributed)
- Servers (Address B)
- Victim (Address A)
- Queries ((spoofed)A → B)
- Responses (B → A)
DNS Response Rate Limiting (RRL)

- Responses rate limited based on:
  - Frequency of incoming domain name/type/source IP
- Responses are small – simply request retry over TCP
- Legitimate clients still have a reasonable chance
- Weaknesses:
  - Relies on a threshold
  - Deals with amplification, but not reflection
DNS Cookies

- Server sends cookie to client
  - Cookie must be included in subsequent requests
  - Server drops requests from clients that don’t have cookies
- Effective for source IP address validation
- Weaknesses:
  - Cannot be effectively enforced
Source Address Filtering: Best Current Practice 38 (BCP38)

- Filter IP packets whose source IP addresses don’t originate in-network
- That’s it!

Queries
((spoofed)A → B)

Attackers
(Globally distributed)

Servers
(Address B)

Victim
(Address A)
Incentives

Solution Effectiveness

BCP38

DDoS Impact

DNS RRL

Increase Resources
We either need to

**incentivize the parties**
capable of effective solutions

or

**develop effective mechanisms**
that can be deployed
by those with incentive
Network Capability Assertion In a Nutshell

• Server enforces source address validation mechanism
  • on demand; or
  • all the time
• To enforce source address validation
  • a server performs a lookup of network capabilities; and
  • ignores requests that don’t validate
Reflection with Enforcement of Source IP Address Validation

Queries ((spoofed)A → B)

Attackers (Globally distributed)

Servers (Address B)

Victim (Address A)
Advertising and Detecting Network Capabilities – in the DNS

- Publish and lookup in .arpa tree in the DNS
  - Example: for 192.0.2.1, query the DNS for 2.0.192.in-addr.arpa
- Network capabilities specified at 8-bit granularity
- Child inherits default policy from parent
- Server assumes defaults until lookup completes