Steps Towards a DoS-resistant Internet Architecture

Mark Handley

Adam Greenhalgh

University College London

Denial-of-Service

- Attacker attempts to prevent the victim from doing any useful work.
  - Flooding Attacks
  - Exploiting Software Weaknesses

- Flooding Attack:
  - Send sufficient traffic to overload network link, router, host, firewall, or any other Internet system.
  - Limited resource can be link capacity, CPU, memory, disk space, quota, or pretty much any other consumable.

Dealing with Flooding

1. Detect flooding attack
2. Ask the network to stop sending you the bad traffic.
3. Attacker's ISP disconnects them.

Internet Service Model

- Single global address space.
- Routers don’t know about flows or applications - just move packets as fast as they can.
- Rely on co-operative end systems to perform congestion control.
- Route advertisement is an “invitation” to send packets, no matter what their purpose.
- Destination-based routing: paths are normally asymmetric.
- Source addresses only used by receiving host.

Threat Model

- Thousands of machines compromised:
  - Rapidly spreading worms
  - Automated scanning by bots
  - Viruses
- Compromised machines used for distributed DoS attacks:
  - Attack traffic can total many gigabits/second.
- Source-address spoofing.
  - Actually not very common because not necessary.
- Reflection attacks
  - Serve as amplifiers
  - Obfuscate attack origin.

Hypothesis

- The Internet Service Model provides many modes of interaction between systems.
  - Some are necessary to do useful work
  - Many are unnecessary, but can be used by attackers.

Question

- Are there cost-effective ways to limit the modes of interaction in such a way that normal traffic is unaffected, and the balance of power moves in favor of defense?

Internet Service Model

- Single global address space.
- Routers don’t know about flows or applications - just move packets as fast as they can.
- Rely on co-operative end systems to perform congestion control.
- Route advertisement is an “invitation” to send packets, no matter what their purpose.
- Destination-based Routing: paths are normally asymmetric.
- Source addresses only used by receiving host.

Internet Service Model

- Single global address space.
- Routers don’t know about flows or applications - just move packets as fast as they can.
- Rely on co-operative end systems to perform congestion control.
- Route advertisement is an “invitation” to send packets, no matter what their purpose.
- Destination-based Routing: paths are normally asymmetric.
- Source addresses only used by receiving host.

Clients and Servers

- To a first approximation, hosts divide into "clients" and "servers".
- Desired service model:
  - Clients can send unsolicited requests to servers.
  - The only traffic that can reach a client is from a server to which it sent a request.

- *Yes there are other things than clients and servers*
  - *We'll get to them later.*

Towards a DoS-resistant Internet Architecture:

Step 1: Separate Address Spaces

- Separate the address space into client addresses and server addresses.
- Allow packets from *client⇨server* and *server⇨client* and nothing else.

Benefits:

- Fast worms prevented (*client⇨server⇨client* is slow)
- Reflection attacks on servers prevented because this needs *server⇨client⇨server* and typically reflectors are “servers”.

This is similar to the asymmetry that NAT creates, but makes it a consistent part of the architecture.

Step 2: Non-Global Client Addresses

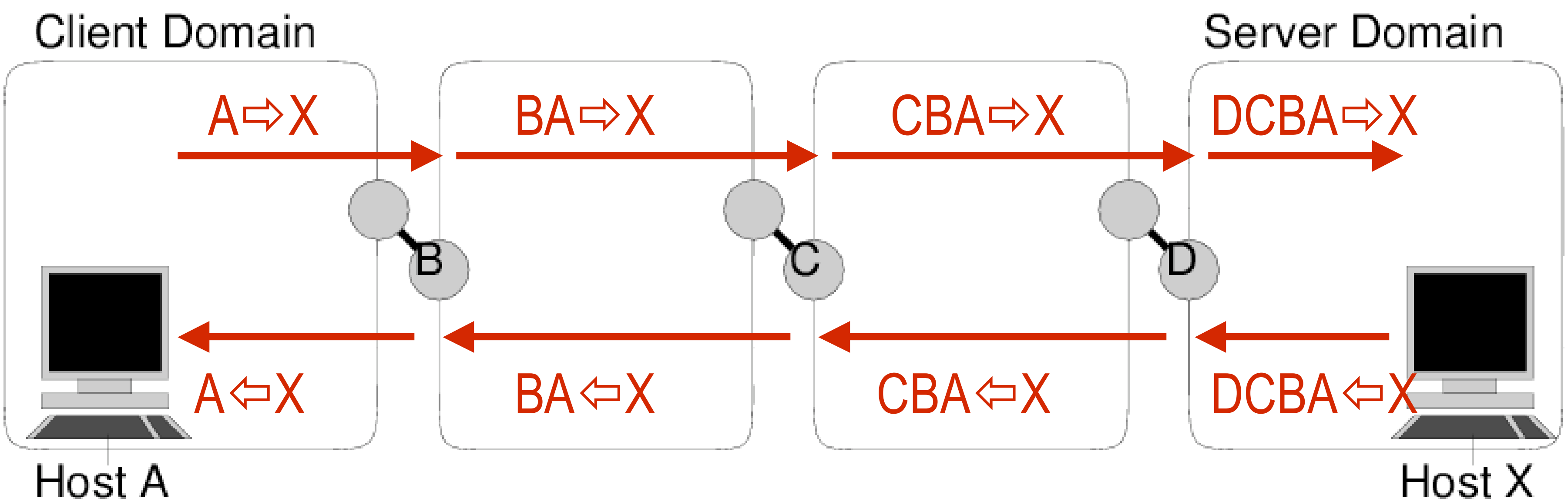
A client address does not need to have global significance.

- Only needs significance on the path back from *server⇨client*
- In fact, a client wants its address to not have global significant, because this prevents *distributed* DoS attacks on a client host.

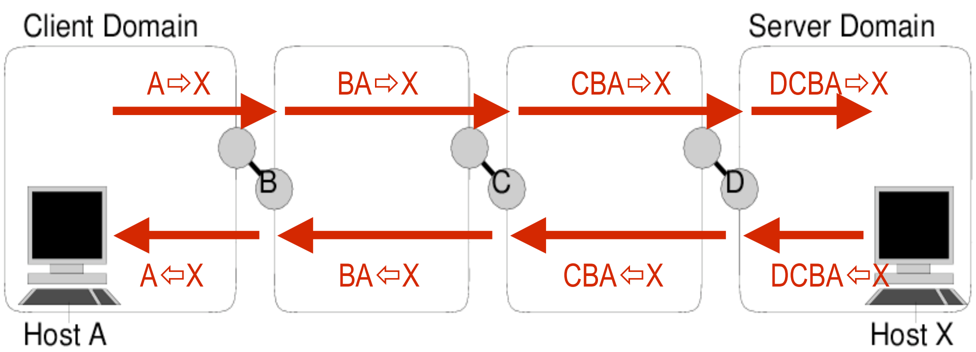
Solution:

- Path-based addressing of clients.

Path-based Addressing



Path-based Addressing



Benefits

- Prevents client address spoofing.
  - Thus reflection attacks on remote clients not possible.
- Prevents DDoS of clients.
  - Client addresses not guessable.
- Paths are symmetric at the inter-domain level
  - Unidirectional traffic is then clearly visible as malicious, even in the core of the Internet.
- Remote subversion of client routing not possible.

Path-Based Addressing Issues

- Client addresses are inherently changable.
  - Needs an additional stable namespace to allow connections to bind to a stable name.
  - HIP would provide one such namespace.

- Routing change can render *server*⇨*client* path unusable if client is idle.
  - Either need *client*⇨*server* keepalives, or client visibility of route changes.

Step 3: Server RPF Checking

- The use of path-based client addresses means routing is symmetric at the inter-domain level.
- This allows all domain boundaries to perform reverse-path forwarding (RPF) checks on *server*⇨*client* traffic.

Benefits

- Server address spoofing is prevented.
- As neither client nor server address can be spoofed, remote injection attacks on ongoing communications (such as TCP Reset injection) are prevented.

Towards a DoS-resistant Internet Architecture:

Step 4: State Setup Bit

- Not all packets are equal. Packets that cause state setup are especially risky from a DoS point of view.
- Introduce a state-setup bit in the IP header.
  - Must be set on packets that cause communication state to be instantiated, and unset on others.
  - Server ignores packets for new flows that don't have bit set.

Benefits

- Protocol-independent way to identify packets requiring special validation.

The State Setup Bit

Benefits

- Stateful firewalls can validate packets with this bit set before instantiating state.
- Server addresses cannot send state-setup packets
  - Routers would drop such packets.
  - State-holding attacks not possible from server addresses.
- Sites might rate-limit state-setup packets.

Inherent conflict between security and network evolution:

- A state setup bit at the IP level makes it easier to evolve transport and application protocols.

Step 5: Nonce Exchange and Puzzles

- Need mechanisms to validate a client, and to add asymmetric costs to communications to change the balance of power towards the server.

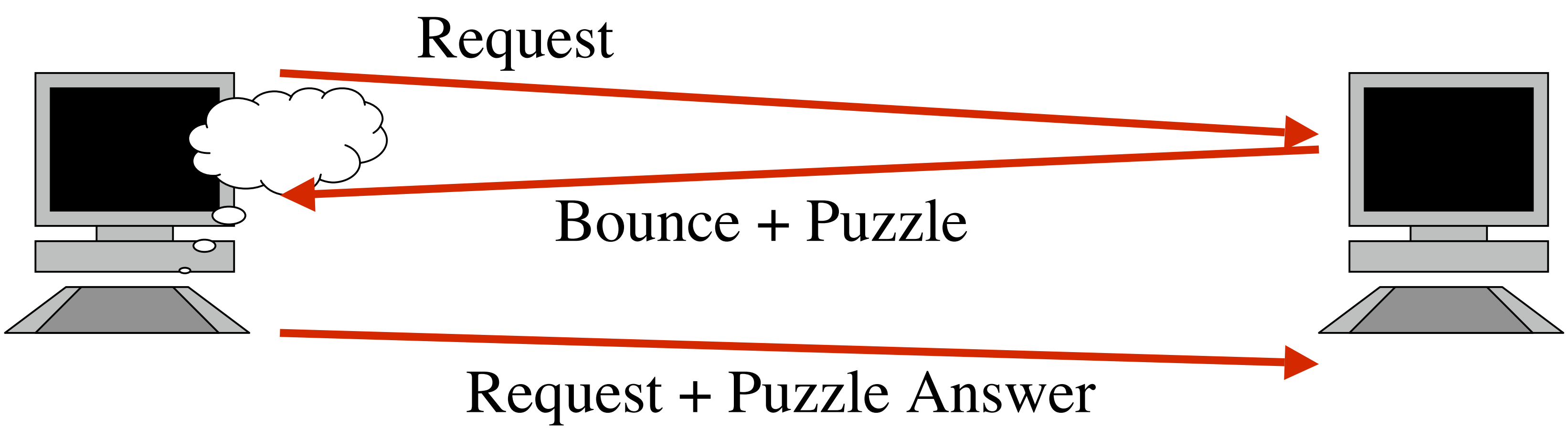
Nonce-exchange:

- Generic response to state-setup packet requiring a nonce to be echoed.

Puzzles:

- Generic response to state-setup packet requiring client solves a puzzle before communication can continue.

Nonce Exchange, Puzzles.



- Not a new idea, but the addressing constraints make it much safer to deploy - much harder to use the puzzle mechanism as a DoS attack in its own right.
- Only helps with IP/Transport level attacks. Application will still need to do DoS prevention.

Towards a DoS-resistant Internet Architecture:

Step 6: Middlewalls

Traditional firewalls are too close to the server host to provide much protection against DoS.

- Need some form of access control that is upstream of the bottleneck link or router.

Middlewall

- A simple special-purpose high-speed firewall deployed in the core the Internet at an inter-domain boundary.
- Performs nonce-validation, issues puzzles, drops specific traffic flows.

Middlewall Activation

- Middlewall normally acts as a transparent relay.
- A middlewall's help is solicited by a destination subnet:
  - For specific sources:
    - Control message travels back along client-address path. Hard to spoof due to RPF checks.
    - Issue puzzles, or do nonce exchange and block specific source.
  - For DDoS attack:
    - General solicitation to issue puzzles, carried in routing messages from destination subnet.
- Interesting question: can a middlewall charge money for the service?

The story so far...

- No rapidly spreading worms.
- No source address spoofing.
- No reflection attacks.
- Clients completely protected from direct attack.
- Servers protected from attack by servers (and clients are much harder to compromise)
- Simple pushback mechanisms against known malicious clients.
- No per-flow state, except when actively solicited by servers.
- Puzzles make all but the largest DDoS attacks unsustainable.
- Large DDoS attacks cannot use unidirectional traffic.
- The remaining attacks mostly look like a flash crowd.

What did we give up?
Potential problems due to loss of symmetry

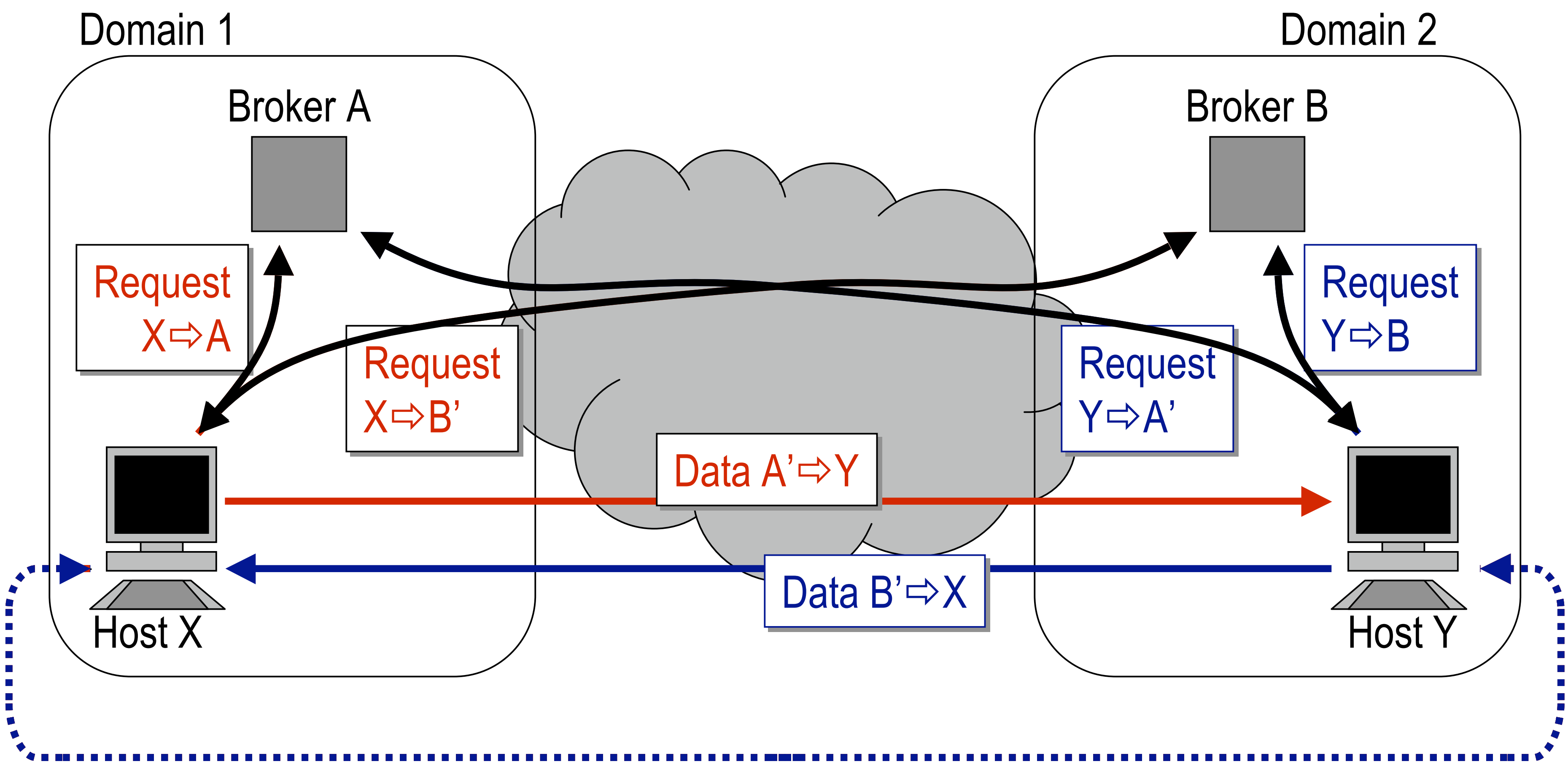
- Application-level relays: SMTP, NNTP, SIP.
  - Need both client and server addresses.
  - As far as possible, try to avoid needing both addresses to be globally routable.

- Peer-to-peer applications and Internet telephony:
  - Need client-to-client communications.

Client-to-client communications.

- Peer-to-peer applications and Internet telephony both have out-of-band signaling/discovery mechanisms which can work client-server.
- The actual client-to-client communication can then be simultaneously setup from both ends.
  - Simultaneous setup is not nearly so vulnerable to DoS because both parties have to consent to it.
  - Needs the help of one or more server addresses to bootstrap - there are multiple possible solutions to this.

Client-to-Client Communication



Summary

- Simple architectural changes can make a big different to the DoS threat space.
- Making asymmetry an integral part of the architecture seems key.
  - “Client” vs “Server” split is a big win.
- Symmetric applications supported through simultaneously setup.
  - More complicated, but not disasterously so.
  - Peer-to-peer may be just too risky though, as it permits fast spreading worms.

Big Question

- What do we actually want from a network architecture?