Firebreak: An IP perimeter defense architecture

Paul Francis
Cornell University
francis@cs.cornell.edu

Firebreak:

A long swath of cleared vegetation used to contain wildfires

Firebreaks can be natural

IP internet amounts to "doorchain" security

- IP packets enter the OS *before* a decision to accept them or not is made!
- A malicious sender can deny you service
- And scan your machine for security holes

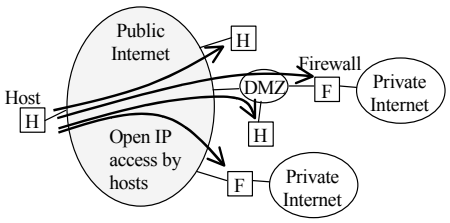
IP internet amounts to "doorchain" security

- IP packets enter the OS *before* a decision to accept them or not is made!
- A malicious sender can deny you service
- And scan your machine for security holes

The apartment doorman is a better model

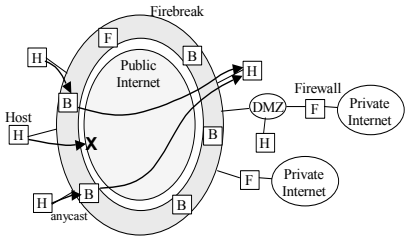
- Accept or reject packets far away from your network interface!
- Including far away from your firewall

Today's IP model is open and anonymous



Today, hosts have open and anonymous IP access to public Internet access points

Firebreak: An IP-level ring of protection



Packets must go through firewall-like boxes near the edge! Destinations control these boxes.

Firebreak model

- Senders cannot send packets directly to receivers
- Rather, packets go through firewall-like boxes ("firebreaks") near the sender
- Receivers can install firewall rules in the firebreak
  - Only rules that pertain to itself, of course

Public and private firebreak usage are very different

- Public destinations (web sites, etc.)
  - "Default" rule is accept all packets
  - If under attack (or overloaded) install filters where needed
- Private destinations (your home PC)
  - "Default" rule is to accept nothing
  - PC installs filters that allow specific sources to initiate specific applications at specific times

Why isn't the Internet built this way???

- The Internet was built by researchers for researchers
  - NOT, contrary to myth, to withstand nuclear holocaust
- Researchers are a trusting and trustworthy lot
  - They didn't think about DDoS, worms, viruses, etc.
  - Plus they had more fundamental issues to deal with

So, how do we get there from here? Can we?

- How to prevent direct IP connectivity for general hosts, but allow it for firebreaks
- How to force packets through firebreaks without changing every edge router
- How to scalably install private filters into the firebreak infrastructure
- How do we identify and authenticate hosts?
- How to make the firebreak itself resistant to DDoS attack
- What is the business model that would drive deployment?

This talk . . .

- . . . explores the feasibility of the firebreak architecture
- By looking at:
  - IP routing issues
  - Firebreak infrastructure issues
    - Host naming and authentication, firebreak filter rules, firebreak infrastructure protection
  - Business model issues
  - Other possible benefits

Two basic IP routing issues

1. How do you force host packets to go to firebreaks . . .
2. . . . while still allowing firebreak packets to get to the intended destination?

Answers:

- IP anycast and router filter configuration

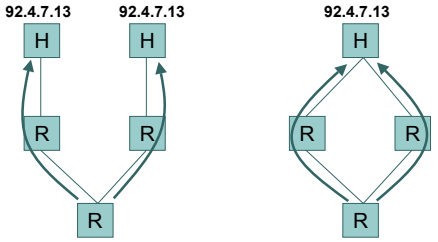
IP anycast is an IP delivery mode

- Many hosts have the same IP address
  - These hosts form the *anycast group*
  - This IP address is the *anycast address*
- A packet sent to the anycast address will go to one of the anycast group members
  - The nearest one
    - by the routers' definition of near

Unicast, multicast, and anycast

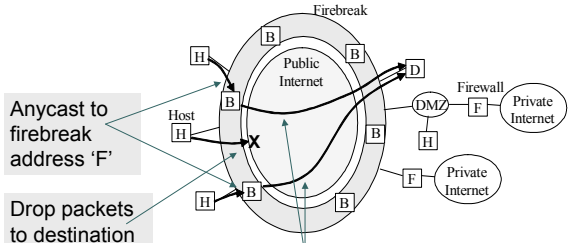
Unicast	One-to-one
Multicast	One-to-many
Anycast	One-to-(one-of-many)

IP anycast requires no router modifications



Routers don't distinguish these two cases:
The same routing algorithms work regardless

Anycast to firebreak, NAT/tunnel to destination

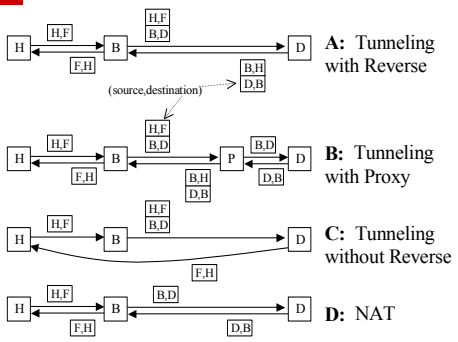


Anycast to firebreak address 'F'

Drop packets to destination address 'D'

Tunnel/NAT packets to destination address 'D'

Various tunneling strategies



A: Tunneling with Reverse

B: Tunneling with Proxy

C: Tunneling without Reverse

D: NAT

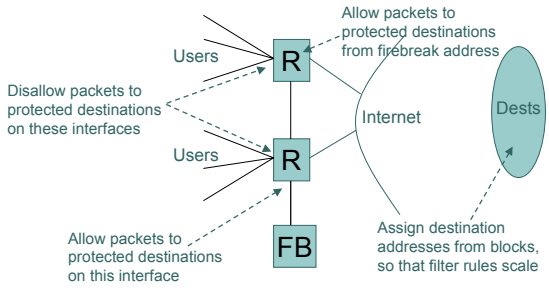
Anycast and router filters

- We more-or-less know how to deploy IP anycast
  - Though much to be learned
- Routers must be configured at two levels, intra-AS and inter-AS

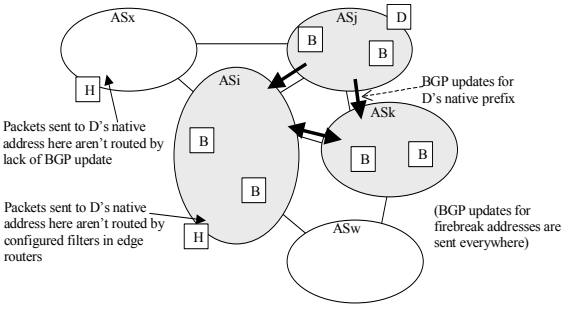
Intra-AS router filters

- Routers in general have rich filtering capabilities
  - Though no doubt at some performance cost, so do it at the edge
- Can block packets towards specified destination addresses
  - From specified source addresses
  - From specified interfaces

Example edge router filtering rules (details will vary)



Firebreak BGP routing example



IP Routing Summary

- Firebreak addresses map into destination addresses
- Sources send to firebreak addresses (i.e. learned from DNS)
  - But not necessary to keep destination addresses secret!
- Edge router filters in participating ISPs
- Don't advertise destination addresses to non-participating ISPs

Some IP routing issues

- Can router filter configuration be made simple?
  - A homogeneous set of rules for all edge routers
  - Destination addresses in a small number of large blocks
- How can we detect errors in the configurations?
  - BGP and router filters
- What do router filters do to performance?
- Can nearby hosts be protected from each other?

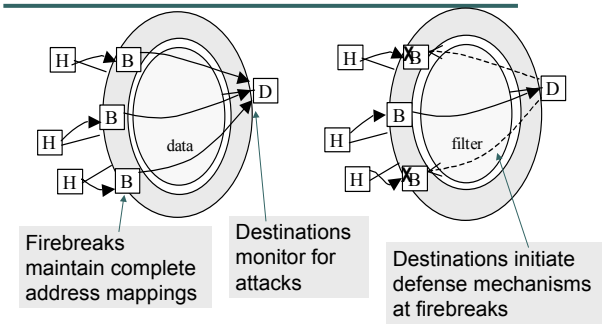
IPv6 would be great here!

- Managing all the (IPv4) destination addresses is hard
  - Router filter rules
  - Firebreak mapping rules
    - Harder still with private usage
- With IPv6, could simply divide address space in half
  - 1-1 mapping from firebreak to destination address
  - Routers filter on one huge block of addresses

Public and private usage very different

- Public usage has a better business model
- Far more private destinations
- Public destinations reachable by default
  - No firebreak rules unless attack
  - Private usage requires perpetual, dynamically changing firebreak rules
- Essentially have to design two different systems!

Public usage architecture



Example firebreak defense mechanisms

- Filter against *likely* source-spoofed packets
  - Note that firebreaks can learn which source address blocks are expected
- Terminate TCP to detect and stop SYN attack
- Fair queue *likely* non-spoofed sources
- Shed some percentage of all traffic

Public usage business model

- Akamai-like company sells DDoS protection service to web sites
- By installing firebreaks in hundreds of locations around the Internet
  - Cooperate closely with ISPs---some profit sharing required
- Indeed Akamai offers DDoS protection services today

Akamai's current DDoS protection approach (believed)

- Origin server IP address kept secret
  - Security through obscurity!
- Two tiers of DNS
  - Dozens (?) of top tier servers, reached by IP anycast. Large TTL.
  - Thousands (?) of second tier servers. Small TTL.
- This is "quite good" protection

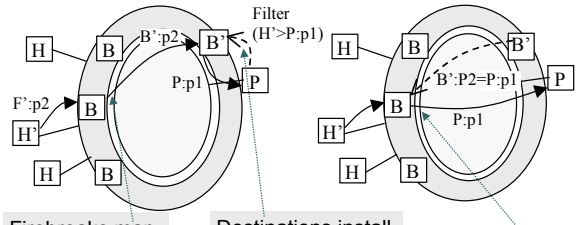
Possible Akamai attacks

- Attack origin servers by discovering IP address(es)
  - "Static" content cached at Akamai proxies ok
  - Akamai could reconfigure those addresses…
- Sustained attack on top tier of DNS
  - But ISPs can traceback attackers and install filters on timescale of hours
  - But if this attack succeeds, *all* Akamai customers are denied service!

Private usage

- When communications is desired, the private host allocates a port and installs a filter in the “firebreak”
  - Filter is specific to a single remote host
- Filter cannot go out to all firebreaks, so require notion of a “home firebreak”

Private usage architecture: “Home Firebreak”



Firebreaks map addresses to home firebreaks (F’ → B’)

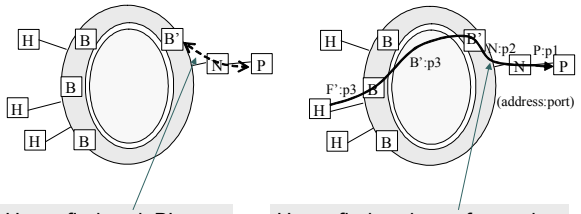
Destinations install filters at “home firebreaks” (B’)

Firebreaks cache address mappings

Mapping firebreak addresses to home firebreak addresses

- One approach:
- Any firebreak may also be a home firebreak
- Each firebreak assigned a range of transport addresses (TA) (addr:port)
- All firebreaks know about all other firebreaks and their assigned TAs
  - 10,000s of firebreaks

Home firebreaks facilitate connectivity through NATs



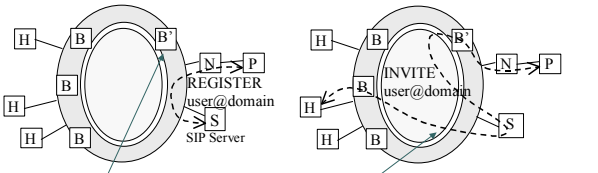
Home firebreak B’ learns NAT-assigned addr and port for P

Home firebreak can forward packets to P via the NAT (or redirect ingress firebreak to P)

Home firebreak issues: setting filters

- Filters cannot always be 5-tuples
  - src/dst IP, src/dst port, protocol
- Because may not know IP address of remote host in advance
- One option: SIP URIs
  - Indeed, use SIP to broker all data communications!
- (SIP = Session Initiation Protocol, designed to do signaling for voice/video)

Usage of SIP to establish data communications



P installs filter (not shown) for SIP server before SIP registration.

SIP invite is filtered by SIP server

After this (not shown), P installs filters for flows negotiated by SIP.

SIP has nice properties for generalized P2P data communications

- Name/identifier that is not topology sensitive
  - Machine and user mobility
- Richer semantics for describing intended application
  - Port number space is limited
  - Can include version numbers, vendor, desired protocol stack (IPsec, SSL), etc.
- User authorization

Private usage business model?

- Not so clear as the public usage business model
  - Home users don't perceive DDoS or port scanning as a threat
- Has a similar problem as IPv6: needs confluence of host and infrastructure capabilities
  - Perhaps if a popular P2P application used it (Kazaa, Xbox, PS2, …)

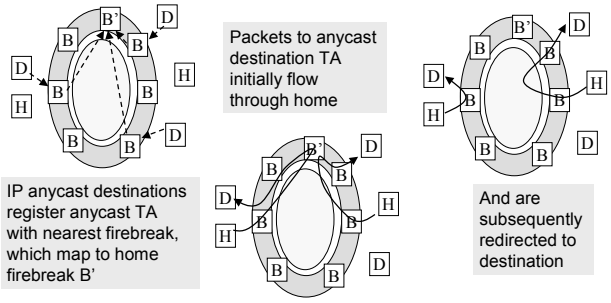
Protecting the firebreak

- Firebreak is useless if it can be attacked!
- Main issue is to prevent resource exhaustion attacks
  - Along the lines of TCP SYN attacks
- Work needed here, but preliminary analysis indicates that the firebreak can protect itself

Beyond firebreaks: generalized IP anycast

- Firebreak model can be extended to provide generalized IP anycast
  - Firebreaks map transport addresses into *one of many* destinations
- Has parallels with Stoica's Internet Indirection Infrastructure (i3)
  - Less general, but more robust and backwards compatible

Generalized IP anycast approach



Generalized IP anycast

- IP anycast long thought to be a powerful tool, but hard to deploy
  - routing protocols, address block allocation, etc.
- Allows any host to become an IP anycast destination without the difficulty of deployment

Project status

- We are building the firebreak boxes and protocols
  - In the linux kernel
- We are building generalized IP anycast
  - Firebreak if an application of this!
- Hope to initially deploy in Internet2