Secure Internet Indirection Infrastructure (I3)

Dan Adkins
UC Berkeley

January 14, 2003

Introduction

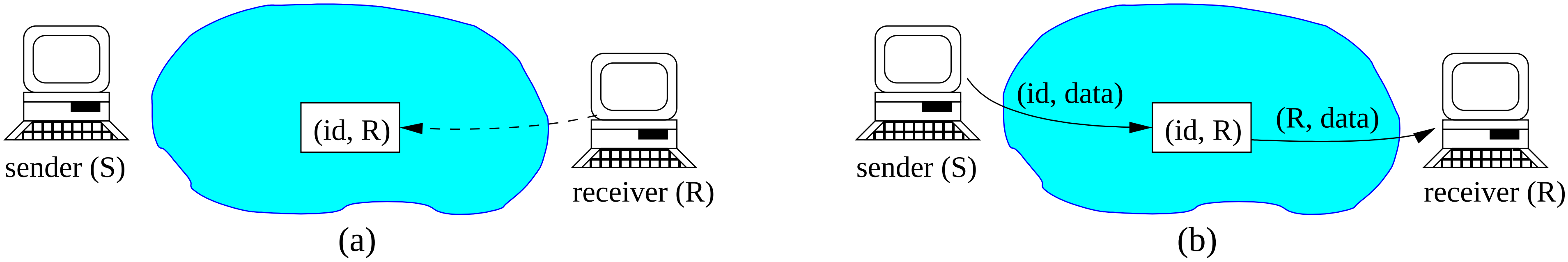
- Internet has two major limitations
  - Flexibility
  - Security
- Recent work addresses flexibility
  - Overlay networks in general
  - I3 in particular
  - Flexibility allows more diverse and powerful applications
  - More control to endhosts can actually increase robustness
- Goal: Network infrastructure that is both flexible and secure
- I3 as a proof of concept

Challenge

- I3 is more vulnerable to malicious attacks than the Internet
  - I3's flexibility is both a feature and a potential for abuse
  - Active networks had this problem
- Can I3 be as secure as the Internet without sacrificing flexibilty?
  - or even more secure?
- We could encrypt everything
  - But that's overkill
  - Only addresses privacy

I3 Overview

- Efficient indirection layer on top of IP
- Rendezvous based communication abstraction (instead of point-to-point)
  - Each packet has an identifier id
  - To receive a packet with identifier id, receiver R maintains a trigger (id,R) in the overlay network



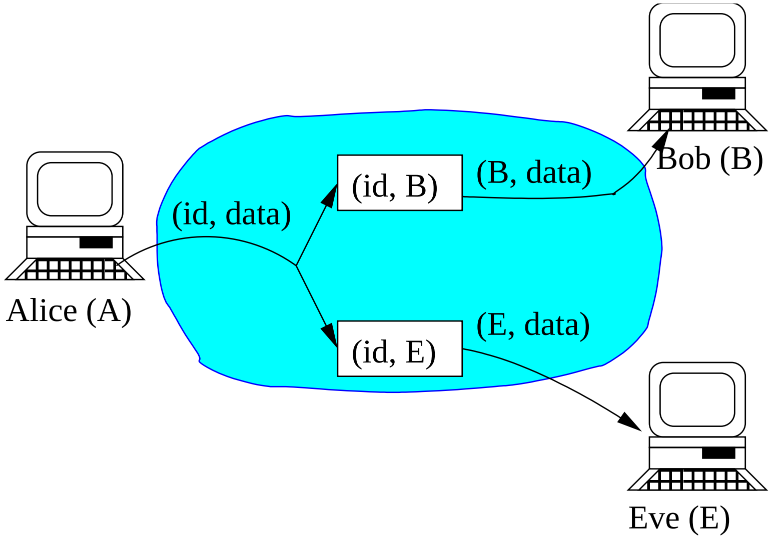
- Triggers consist of (id, dest)
  - dest can be either ID or IP address
  - Multiple triggers with same ID and trees of triggers possible

Problem statement

- Want to
  - Avoid eavesdropping
  - Avoid impersonation
  - Avoid DoS
    - ∗ on infrastructure: loops, confluences
    - ∗ on clients: reflection
- Without losing flexibility
  - Trees of triggers
  - Ability to choose ID's
    - ∗ Place triggers on specific servers
  - Service composition
- With little overhead

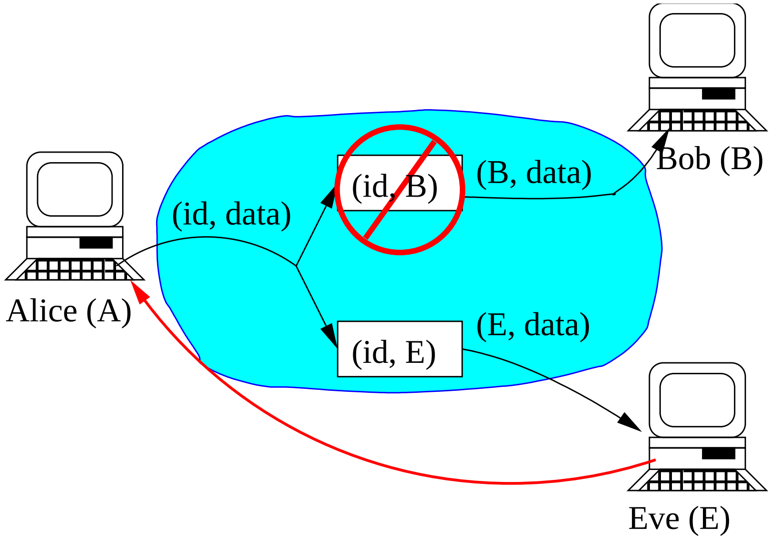
Eavesdropping

- Eve wants to listen to Alice and Bob's traffic
- Eve inserts trigger with same ID as Bob's trigger
  - Possible as a consequence of multicast
- Undetectable to Alice or Bob
- Unavoidable if Bob's trigger is public



Impersonation

- Active version of eavesdropping
- Eve impersonates Bob to Alice
- Eve takes over Bob's public trigger when it expires
  - due to crash, DoS, network outage, etc.

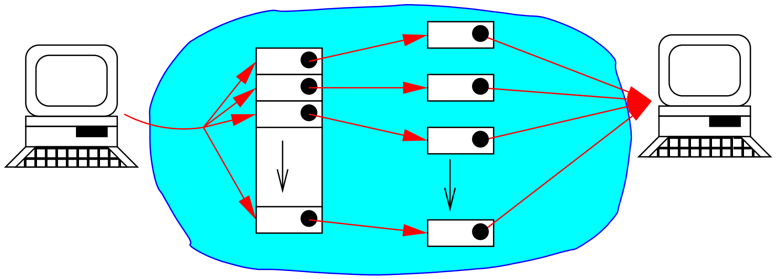


Loops and confluences

- Some troublesome topologies can lead to DoS

- Loops
  - May be formed maliciously or inadvertently
  - Causes an endless stream of packets

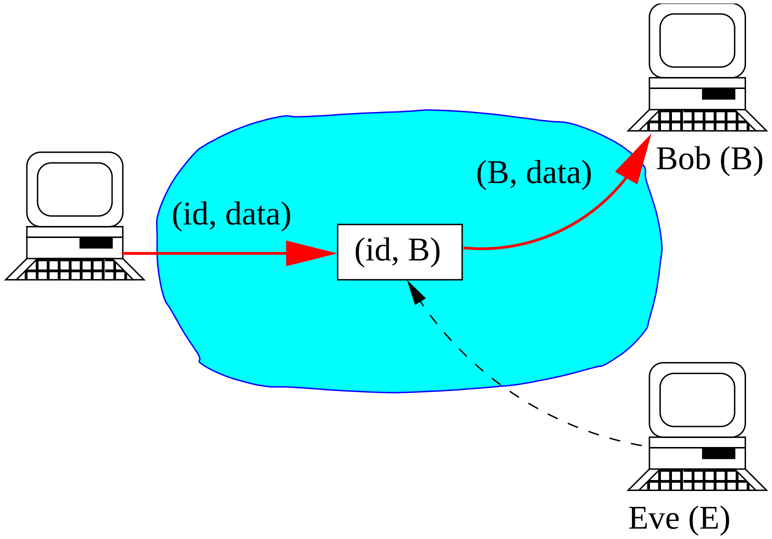
- Confluences
  - Tree expanding out then in
  - Can be used as a packet multiplier
  - Roughly speaking, any unwanted convergence of paths



Reflection

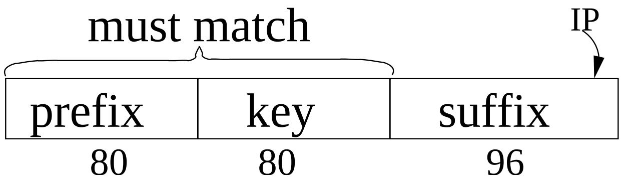
- Eve subscribes Bob to high volume traffic

- An attacker must be able to insert a trigger on the victim's behalf



Solution: constrain triggers

- Idea: maybe arbitrary triggers aren't necessary
  - (x,y) such that x and y are independent
- Only allow trigger (x,y) if x=G(y) or y=H(x)
  - where G and H are one-way hash functions
  - I3 identifer changes:



  - Actually, x.key=G(y.key), so end-hosts have some choice
- Servers will check constraints
- Solves eavesdropping, loops, confluences (?)
- *Preventive solution*

Problems

- Eavesdropping
- Impersonation
- Loops
- Confluences
- Reflection

Eavesdropping

- Insert trigger (G(y),y)
  - G(y) is a public ID
  - Attacker must invert G to insert trigger
- y can be an ID or IP address
- y.key must be kept secret
  - Ok to send trigger insertion message in the clear
  - If Eve can snoop trigger insertion, Eve already has local network access
    - No worse than Internet
- What if Eve inserts (x,H(x)) where x=G(y)?
  - Triggers of form (G(y),y) always take precedence

Problems

- Eavesdropping
- Impersonation
- Loops
- Confluences
- Reflection

Loops and confluences

- Triggers can be either (G(y),y) or (x,H(x))
  - (G(y),y) — tree built from receiver
  - (x,H(x)) — tree built from sender
- Nearly impossible to form a loop with constrained triggers
  - Requires finding hash chain that eats itself
  - As hard as inverting one-way function
- Confluences on a single ID are impossible too
  - Can only build trees from sender or receiver
  - No way to connect them without inverting G or H
- *But, confluences on I3 nodes are still possible!*

Server confluence

- DoS against infrastructure still possible
  - Attacker can overload I3 node by directing confluence towards *multiple* ID's on the same server
  - Not technically a confluence (no convergence point)

- Use push-back
  - I3 servers or clients under load may remove triggers
    - Weighted fair queueing helps identify which triggers to remove
  - Dead end triggers are a problem for the infrastructure in general
  - Solution: When a packet arrives that matches no trigger, send it back
  - The sender (another I3 server) should remove the trigger which caused the packet

- Push-back is a good idea in general for error detection

- Push-back is more effective if each host connects to a nearby I3 server.

Problems

- Eavesdropping
- Impersonation
- Loops
- Confluences
- Reflection

Impersonation and Reflection

- Impersonation
  - Only a problem when server goes down
  - If you really care, exchange secrets or certificates
- Reflection
  - Principle: You should only receive packets which you (implicitly) request
  - Solution: Challenges
    - Trigger insertion pointing to an IP address must come from that address
    - Server sends a challenge to that address

Tradeoffs

- Overhead of checking constraints
  - Trigger insertion increased from 19us to 24us
- Challenges cost an extra RTT
- True service composition breaks
  - Requires per-flow state
  - We have solution with arbitrary triggers which won't impact service of constrained triggers
    - Constrained triggers have precedence over arbitrary triggers
    - But, arbitrary triggers require higher overhead checks

Conclusion

- I3 can be flexible without compromising security and performance
  - *with constrained triggers, not worse than today's Internet*

- End-hosts can use I3's flexibility to improve resilience against various attacks