UCLA

SAVE: Source Address Validity Enforcement Protocol

Jun Li, Jelena Mirkovic, Mengqiu Wang,
Peter Reiher and Lixia Zhang

UCLA Computer Science Dept
10/04/2001
{lijun, sunshine, wangmq, reiher, lixia}@cs.ucla.edu

UCLA

Motivation

- Provide routers with information on the valid incoming interface for a given source address
- Filter out packets with invalid source addresses
- Would be helpful for
  - Many security issues
  - Building multicast trees
  - Network problem debugging
  - Services relying on accurate source addresses ...

UCLA

Outline

- Motivation
- The Idea
- Handling Routing Changes
- Security and Deployment
- Simulation and Implementation
- Related Work
- Ongoing Work
- Conclusions

UCLA

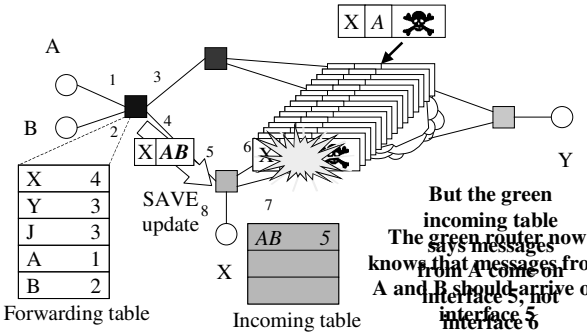
The Idea

- Build an incoming table at a router that specifies valid incoming interfaces for address spaces
  - Cannot be derived from forwarding table due to routing asymmetry
  - Cannot be designed by reversing routing protocol
    - Should be designed to inform routers about the path that has ALREADY been chosen
  - Cannot augment routing updates to carry SAVE info
- So, how?

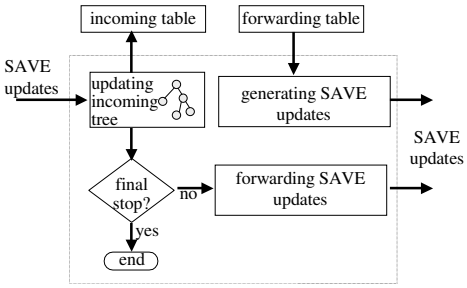
Desired Properties of SAVE

- Routing protocol independence
- Immediate response to routing changes
- Security
- Incremental deployment
- Low overhead

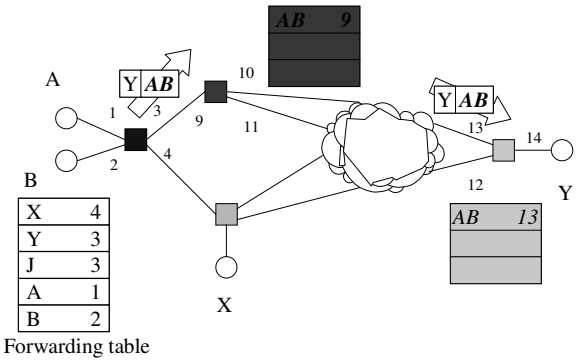
Example



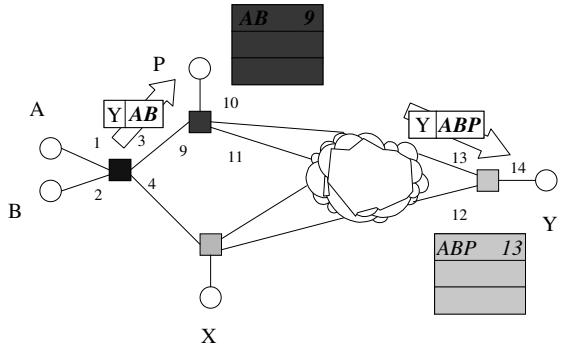
Architecture



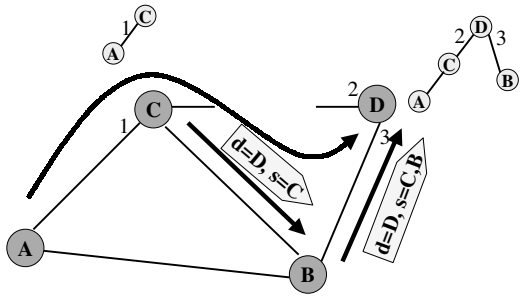
Example



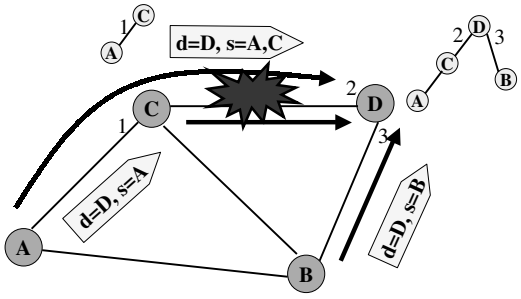
Example



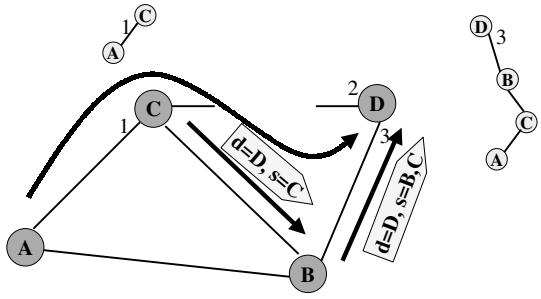
Handling Routing Changes



Handling Routing Changes



Handling Routing Changes



UCLA

Security of SAVE itself

- Essentially the same problem as securing routing protocol
- Requirements
  - SAVE updates must only be exchanged between routers, excluding hosts
  - Trust relationship between routers must be established beforehand
  - SAVE updates must be signed or encrypted
  - Processing of SAVE updates must be lightweight

UCLA

Simulation

- All routers run SAVE protocol + routing protocols
- Transit-stub topology generated using GT-ITM
- BGP as inter-domain routing protocol
- RIP as intra-domain routing protocol
- Some asymmetric routes

UCLA

Deployment

- Can only be incremental
  - Have to deal with legacy routers
  - Incoming table will not cover the whole Internet
  - Deployment at different location has different impact
- Some real issues
  - Mobile IP
  - Tunnelling
  - Multipath routing
  - .....

UCLA

Simulation Goals

- Effectiveness – are all spoofing packets successfully detected and dropped?
- Correctness – are some valid packets dropped erroneously?
- Transient behavior
- Cost

Effectiveness and Correctness

- Each packet source generates both valid and spoofing packets
- Spoofing source addresses randomly chosen from a pool of all source addresses in the network
- Every router is under an average load condition
- Results:
  - In all scenarios all spoofing packets were detected and dropped
  - Without routing changes no valid packets were dropped

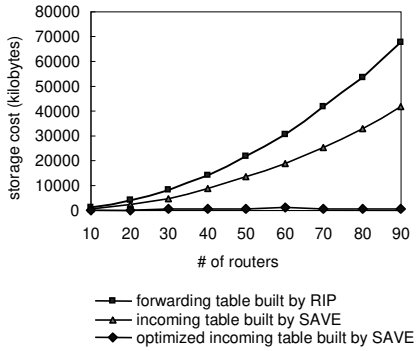
Cost of SAVE

- Compared cost of SAVE with costs of routing protocols (BGP and RIP)
- Bandwidth cost:
  - compared bandwidth consumed by SAVE updates with that consumed by routing updates
- Storage cost:
  - compared the size of incoming table with the size of forwarding table

Transient Behavior

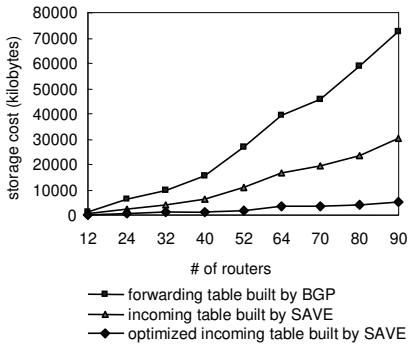
- Route changes introduce a transient period for SAVE to adjust every incoming table along the new route
- During this period valid packets can be dropped on new route
- Assuming that SAVE packets have same propagation delay as data packets, inconsistency occurs if:
  - data packet is sent out on new route BEFORE new SAVE update validating this route
  - data packet is filtered at a router on the path BEFORE new SAVE update is processed

Storage Cost (single domain)



UCLA

Storage Cost (multiple domains)



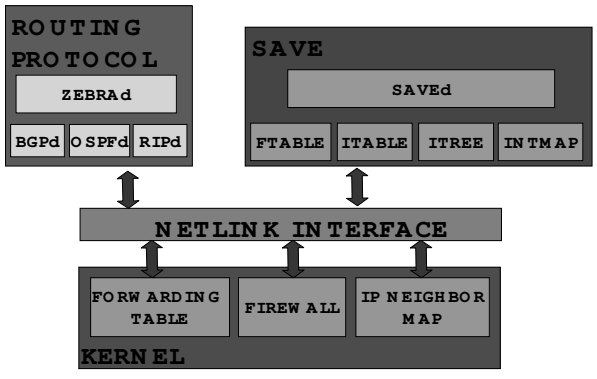
UCLA

Related Work

- Cryptographic Methods
  - High computation overhead of cryptographic operations
- Forwarding-table-based filtering
  - Routing asymmetry leads to erroneous packet dropping

UCLA

Implementation in Linux



UCLA

Related Work

- Ingress and egress filtering
  - Very ineffective if partially deployed
- Packet tracing
  - Complex and expensive
  - Ineffective when the volume of attack traffic is small or the attack is distributed
  - Reactive, not preventive

UCLA

On-going Work

- Cooperation with Purdue University on partial deployment investigation
- Implementation
  - IXP router implementation
  - Cisco router implementation
- Testing
  - FTN testbed
  - Utah testbed
- IETF draft

UCLA

Conclusions

- Filtering improperly addressed packets is worthwhile
- Asymmetric network routing demands an incoming table
- SAVE builds incoming tables correctly with low bandwidth and storage overhead
- SAVE has demonstrated its correctness and effectiveness through simulation
- Implementation is under way