SIPD: a practical SDN-based IP spoofing defense method

Chen Li, Yu Ding, Tongxin Li, Jun Li, Xinhui Han
Peking University and University of Oregon

Background

- By sending packets with fake source IP address, attackers can
  - Hide themselves
  - Pretend to be someone else
  - Hijack a established connection
  - Flood the victim by using public services as reflector

How to defense?

Existing methods…

- Do not allow spoofed packets to get out of sub-network

Need all ASes to deploy filtering rules

Attackers from vulnerable subnets can fake any IP

- Routers add fingerprints into passing by packets, end-hosts perform detection

Require plenty of special routers in the network

End-hosts need to keep identity data

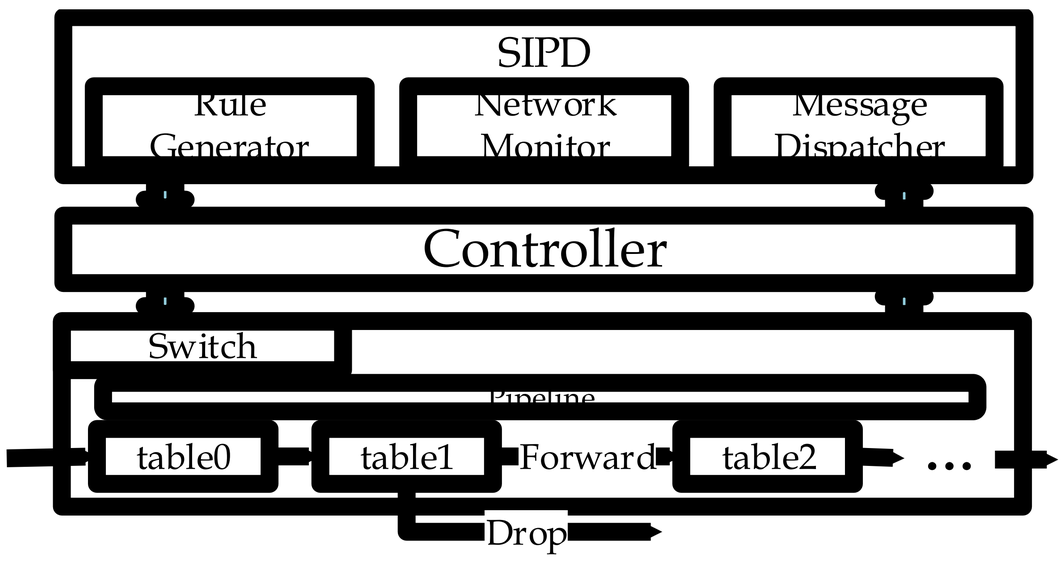
- Filter spoofed packets based on hop-count from sender

High false-positives and false-negatives

Attackers can fake initial TTL to bypass detection

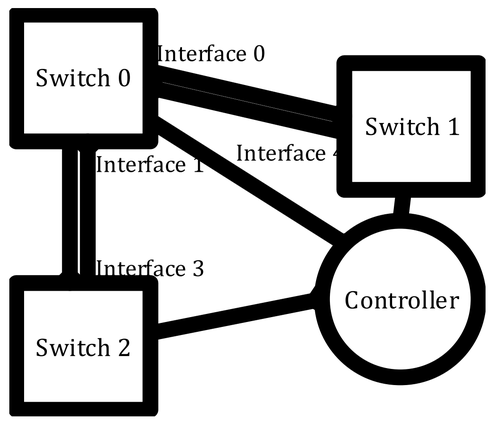
We need to use a better way to solve the problem

System Architecture



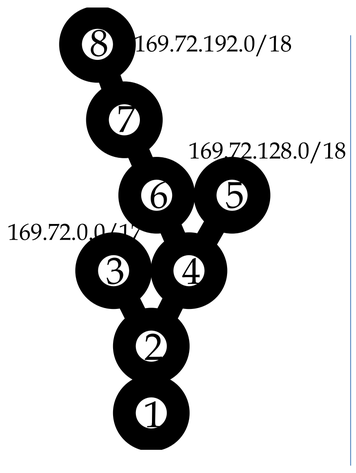
- Automatically discover topology and generate filtering rules
- Quickly respond to network changes
- Enable cooperation between SDNs, and can be incrementally deployed

Topology discovery



- Make switches send out special LLDP packet at all interfaces
- After received the packet, switch sends a report to the controller
- Controller figures out the topology of network

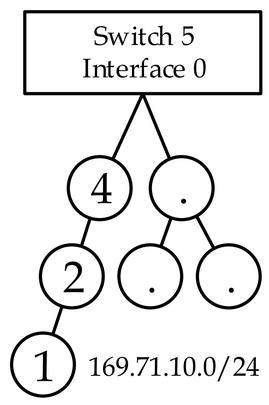
Build forwarding tree



- For each switch, explore the path from it to every destination address space in its forwarding table
- Expand the tree nodes when destination space splits into several sub-spaces
- Follow until the path reaches destination or border switch
- Build forwarding tree as an abstract of forwarding path

Figure: forwarding tree for target 169.72.0.0/16 from switch 1

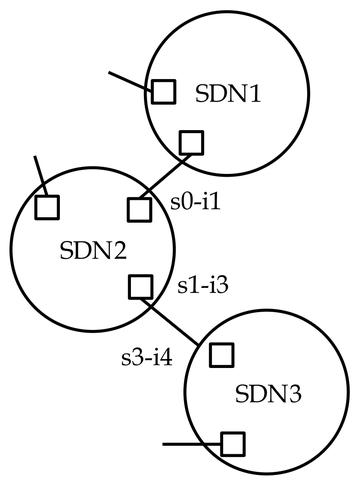
Build incoming tree and generate rules



- Controller collects information from all related forwarding trees, extract the source address spaces that may come to each interface of each switch
- Form a hierarchical relationship based on forwarding path
- Represent all source address spaces that should come in from specified interfaces

- Combine the nodes of incoming trees to get filtering rules
- Figure: incoming tree of switch 5's interface 0

Between SDNs



- Controller collects source address spaces and sends it out from all border switches
- Once a border switch gets message, it forwards the message to the controller
- Controller applies filtering rules at all border switches
- If the destination address space in message is not within its domain, the controller send the message to its destination, and duplicate it when needed

- The controller sends out the message both regularly and when the network change affect the border switch

For more information, please contact Chen Li(icst-lichen@pku.edu.cn)