Source Address Validation: from the Current Network Architecture to SDN-based Architecture

Jun Bi
Tsinghua University/CERNET
GFI 2013
Nov. 20, 2013
Content

• Source Address Validation Architecture (SAVA)
  – SAVA solutions
  – SAVA Implementations
  – SAVA deployment at CNGI-CERNET2

• Leveraging SDN to enhance Source Address Validation
  – Access: Software Defined SAVI
  – Intra-AS: SDN based CPF
  – Inter-AS: Collaborative On-demand Spoofing Defense

• Conclusion
Source Address Validation Architecture (SAVA)
Source Address Spoofing

• Source address spoofing still a problem
  – Arbor Network annual network security report
  – MIT spoofer project
  – NANOG discussions

• Tsinghua university / CERNET proposed:
  – Source address validation architecture (SAVA) and solutions for IPv6
  – Solutions implemented, collaborating with domestic vendors
  – Deployed at CNGI-CERENT2 backbone and 100 universities’ campus networks
  – Co-funders of IETF SAVI WG
    – RFC 5210 SAVA
    – RFC 7039 SAVI Framework
SAVA: Source Address Validation Architecture (RFC 5210)
Access Level: SAVI-CPS
(Control Packets Snooping based SAVI)

SAVI-DHCP

SAVI-Stateless

SAVI-MIX
SAVI-CPS Implementation

• Huawei
• ZTE
• H3C
• Ruijing
• Digitial China
• Centac
• Bitwa
• L3/L2 switch, WLAN
Intra-AS Level: CPF (Calculation based Path Forwarding)

- Key problem to resolve
  - Reduce false positive caused by uRPF at partial deployment

**Path Table**

Src \(\rightarrow\) (inIf1)A1 \(\rightarrow\) (inIf2)A2 \(\rightarrow\) ...... \(\rightarrow\) (inIfn)Dst

**FTDB**

Src, Dst, inIF

**FIT**

Fit: Flow Info Table

Telnet+Scripts configure ACL
CPF Implementation

Topology mgmt  L3 devices mgmt

Spoofing alarm  SAVI user mgmt
Inter-AS Level: SMA
(State Machine based Authentication)

- Key problem to resolve
  - Incentive for deployment
- Trust Alliance
- ACS
  - Each member AS has a control server to negotiate parameter of SMs of each peer to trigger the same tag (random number) sequence
- ASBR
  - Add tags in IPv6 packets (in option header) and validate tags in destination
- Incentive
  - source address of Each AS can’t be spoofed within the Alliance
SMA Implementation

• Huawei NE40E core router line cards with 10G, GE ports
• Bitway BE12000 core router line cards with OC48, GE ports
• Centec special box with 10G, GE (co-located with legacy routers)
SAVA Deployment at CNGI-CERNET2
SAVA Deployment at CNGI-CERNET2
Leveraging SDN to enhance Source Address Validation
Motivations: enhancing Access

• Complex scenarios
  – Address assignment methods: Manual, SLACC, DHCP, SEND, Mixed, …
  – Access methods: LAN, WLAN, DSL, 3G, …
  – Mobility: local, across-network
  – Special cases: IPv6 transition, DNA, … addr. related new stuff
  – Solutions implemented at switches for all scenarios
    • Complex for design and implementation
    • Low efficiency (most scenarios are not common cases)

• Complex configuration
  – Coherent configurations for ALL switch ports at SAVI “perimeter” in the whole access network

• Can we migrate complexity from switch to server ?
Software Defined SAVI (SDN-SAVI)
Motivations: enhancing Intra-AS

- Complexity to deal with legacy management and control interfaces
  - No unified MIBs (private MIBs)
  - No unified sampling protocol (sFlow, NetFlow, NetStream, etc)
  - No friendly programmable interfaces to configure ACLs (telnet + scripts are dangerous for production nets)

- Performance
  - Delay of network status update
  - Delay of control update

- Require unified and realtime mgmt/ctrl interfaces
SDN based CPF

CPF

Global view

SNMP xFlow TCL telnet

NOS (Network OS)

Control Plane

OpenFlow Protocol

Data Plane

Network APP
Filtering Rules Enforcement Delay causes false positive, reflects the filtering effect.
Demonstrated at INFOCOM2012

- **InSAVO: Intra-AS IPv6 Source Address Validation Solution with OpenRouter**

**Central Control:** To get a global forwarding path and then resolve false positive of filtering information caused by asymmetric routing more than ingress filtering.

**Integrated Protocol:** To take place of SNMP/xFlow/Telnet in order to reduce the complication caused by multiple control interfaces with OpenFlow protocol.

**Evolvable Deployment:** To provide software-defined abilities by extending OpenFlow, but also give a tradeoff between existing hardware and evolution cost.
Motivations: enhancing Inter-AS

- Security: SMA uses lightweight tags for verification. An attacker might monitor packets in the backbone and replay the tag with spoofed packets.
  - Solution: cryptographic tags to prevent replay
- Cost: Per-packet crypto marking incurs heavy data processing overhead beyond hardware capacity.
  - Solution: on-demand defense to reduce overhead
  - “When”: defend only attack is detected (cost effective)
  - “Which”: define $N$ defense functions, chosen by operators by the type of attack
  - “Who”: only filtering the specific flows with self-benefit
CoFence: Collaborative On-demand Spoofing Defense

• Distributed inter-AS collaboration
  – Deployer discovery, peering and defense invocation

• Spoofing defense functions
  – Against d-DoS: DP and CDP (CDP uses crypto marking)
  – Against s-DoS: SP and CSP (CSP uses crypto marking)
  – Extensible: can define more functions

• Function invocation
  – Quadruple: \((function, parameters, prefix, time)\)
    • Function: the function to be invoked
    • Parameters: parameters for the function (e.g., keys for crypto)
    • Prefix: the src/dst prefix to be protected
    • Time: the time duration for this invocation

• These lead to SDN-based design
SDN (not OpenFlow) based CoFence

Higher-layer features can be added in apps, e.g., attack monitoring, traffic analysis, AS-wide policy and control, manual-config GUI and auto-config APIs for IDS.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

CoFence-d communicates with controller and manages tables to define data-plane behavior.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.

Controller communicates with other domains, and bridges apps with routers, using SDN interfaces. It provides CoFence-specific functionalities and maintains domain-wide information.
“WE-Bridge” proposed in *FINE*

- *WE-Bridge* proposed in China 863 High-tech R&D project *FINE*
- Demoed at CANS13 and SuperComputing2013
Conclusion

- SAVA and solutions
  - Architecture
  - Access level: SAVI-CPS
  - Intra-AS level: CPF
  - Inter-AS level: SMA
  - Implementation, and deployment at CNGI-CERNET2
- Leveraging SDN to enhance Source Address Validation
  - Motivations
    - Handling complexity
    - Providing agility
    - Improving performance
  - Programmability is key to decoupling infrastructure and functionality (to migrate the complexity to APPs)
  - Leveraging centralized view for access (e.g. configuration, mobility) and intra-AS, and negotiated view for inter-AS SAV
Thanks!