Source Address Validation: Problem of Existing Solutions

draft-li-sava-intra-domain-use-cases-00

Dan Li (Tsinghua)
Jianping Wu (Tsinghua)
Yunan Gu (Huawei)
Lancheng Qin (Tsinghua)
Tao Lin (H3C)
Source Address Validation (SAV)

- The traditional Internet architecture lacks the validation of a packet’s source address.

- Source address spoofing is dangerous.
  - Well documented in RFC 6959

- SAV is important to prevent source address spoofing.
Existing SAV Solutions

- **Host-level SAV**
  - ✓ SAVI [RFC 7039]
    - ◆ Problem: Requires all the access networks (sub-nets) to deploy simultaneously

- **Network-level SAV**
  - ✓ Ingress ACL [RFC 2827]
    - ◆ Problem: Requires manual configuration to update
  - ✓ uRPF
    - ◆ Strict uRPF [RFC 3704]
    - ◆ Loose uRPF [RFC 3704]
    - ◆ Feasible-Path uRPF (FP-uRPF) [RFC 3704]
    - ◆ Enhanced Feasible-Path uRPF (EFP-uRPF) [RFC 8704]
Strict uRPF and the Problem

- Take the source address as a destination address to lookup the FIB.
- If the outgoing interface of the FIB matches the incoming interface of the packet, then pass

**FIB at Router 3:**
- Prefix: P1
- NH: interface 1
- Prefix: P2
- NH: interface 2

Flow 1 with source address P1 is correctly accepted at interface 1
Flow 2 with source address P2 is incorrectly denied at interface 1
Flow 3 with source address P2 is correctly accepted at interface 2
Loose uRPF and the Problem

- Take the source address as a destination address to lookup the FIB
- If the address exists in the FIB, then pass

FIB:
Prefix: P1
NH: interface 1
Prefix: P2
NH: interface 2
Prefix: P3
NH: interface 3

Flow 1 with source address P1 is correctly accepted at interface 1
Flow 2 with source address P2 is correctly accepted at interface 1
Flow 3 with source address P3 is incorrectly accepted at interface 1
FP-uRPF and the Problem

- Take the source address as a destination address to lookup the RIB (including other routing information besides FIB)
- If the outgoing interface of the RIB matches the incoming interface of the packet, then pass

Flow 1 with source address P1 is correctly accepted at interface 1
Flow 2 with source address P2 is correctly accepted at interface 1
Flow 3 with source address P1 is incorrectly denied at interface 2
EFP-uRPF Algorithm A

- EFP-uRPF is designed for Inter-AS case
- Set all the prefixes received for an AS on each customer interface that received an update

Flow 1 with source address P1 is correctly accepted at interface 1
Flow 2 with source address P2 is correctly accepted at interface 1
Flow 3 with source address P3 is correctly accepted at interface 1
EFP-uRPF is designed for Inter-AS case
Set all the prefixes received for an AS on each customer interface that received an update

Flow 1 with source address P1 is incorrectly denied at interface 1
Flow 2 with source address P2 is incorrectly denied at interface 1
Flow 3 with source address P3 is incorrectly denied at interface 1
EFP-uRPF Algorithm B

1. Set $I = \{\text{interface 1}, \text{interface 2}\}$
2. $P = \{P1, P2\}$
3. $A = \{\text{AS1}\}$
4. $Q = \{P3\}$
5. $Z = \{P1, P2, P3\}$ for interface 1 and interface 2

- Flow with source address in P1 is correctly accepted at interface 1
- Flow with source address in P2 is correctly accepted at interface 1
- Flow with source address in P3 is correctly accepted at interface 1
Cases When All uRPF Solutions cannot Work
Case 1: Inter-AS

BGP:
- Prefix: P1
  - NH: Router1
- Prefix: P2
  - NH: Router 1
- Prefix: P3
  - NH: Router 3

AD1
Multiple ASes

AS1
- Router2
  - [P1,P2]
  - [P3]

AS2
- Router3
  - [P3]

AD2
AS3
- Router1
  - [P1,P2]

AD3
AS4
- Router5
  - [P3]

Strict uRPF drops the legitimate packet
Loose uRPF accepts the legitimate packet
Feasible- path uRPF drops the legitimate packet
EFP-uRPF A drops the legitimate packet
EFP-uRPF B accepts the legitimate packet

- drops the forged packet
- accepts the forged packet

Feasible- path uRPF drops the legitimate packet
EFP-uRPF A drops the legitimate packet
EFP-uRPF B accepts the legitimate packet

Stric
Case 2: Intra-AS

**AS**

**Static:**
Prefix: P1
NH: Router1

**IGP:**
Prefix: P2
NH: Router 3
Prefix: P3
NH: Router 3

**Prefixes:**
- P1
- P2
- P3

**Routing Paths:**
- Static: Prefix: P1 NH: Router1
- IGP: Prefix: P2 NH: Router 3 Prefix: P3 NH: Router 3

**Routers:**
- Router1
- Router2
- Router3
- Router4
- Router5

**Prefixes:**
- [P1]
- [P2, P3]

**uRPF Policies:**
- **Strict uRPF** drops the legitimate packet
- **Loose uRPF** accepts the legitimate packet
- **Feasible-path uRPF** drops the legitimate packet
- **EFP-uRPF** does not apply at the intra-AS case

- **Prefixes:**
  - P1
  - P2
  - P3

- **Routing Paths:**
  - Static: Prefix: P1 NH: Router1
  - IGP: Prefix: P2 NH: Router 3 Prefix: P3 NH: Router 3

- **Routers:**
  - Router1
  - Router2
  - Router3
  - Router4
  - Router5

- **Prefixes:**
  - [P1]
  - [P2, P3]

- **uRPF Policies:**
  - **Strict uRPF:** drops the legitimate packet
  - **Loose uRPF:** accepts the legitimate packet
  - **Feasible-path uRPF:** drops the legitimate packet
  - **EFP-uRPF:** does not apply at the intra-AS case

- **Prefixes:**
  - P1
  - P2
  - P3

- **Routing Paths:**
  - Static: Prefix: P1 NH: Router1
  - IGP: Prefix: P2 NH: Router 3 Prefix: P3 NH: Router 3
Thanks!
Any comments?
Use Case 3: Inter-AS

BGP:
Prefix: P1
NH: Router 1
Prefix: P2
NH: Router 3
Prefix: P3
NH: Router 3

BGP:
Prefix: P1
NH: Router 2
Prefix: P2
NH: Router 1
Prefix: P3
NH: Router 5

Strict uRPF drops the legitimate packet
Loose uRPF accepts the legitimate packet
Feasible-path uRPF drops the legitimate packet
EFP-uRPF does not mention this case