# Peering Trends & Best Practices

European Peering Forum 2024

Florian Hibler <florian@arista.com>
Patrick Prangl <pprangl@arista.com>

# Why should you trust me?

**Patrick Prangl**

Advanced Services Engineer

- Organically grown in Austria
  - Not the one with the Kangaroos
- 10+ years operational experience with Service Providers
- Design and architecture for large scale SPs, IXPs and Data Centers across EMEA

# IPv4 over IPv6 networks

RFC5549/8950

# What is the problem?

- IPv4 addresses are expensive
  - Most efficient use of your existing IPv4 addresses
  - Sourcing IPv4 addresses for P-t-P connections is a difficult business case
  - Drives ‘cost of product’ up

- Growing pains
  - Renumbering at IXPs are causing pain for the peering ecosystem

- No more IPv4 address ‘re-using’ for ‘PtP transit networks’

# IPv6 to the rescue

Network Working Group
Request for Comments: 5549
Category: Standards Track

F. Le Faucheur
E. Rosen
Cisco Systems
May 2009

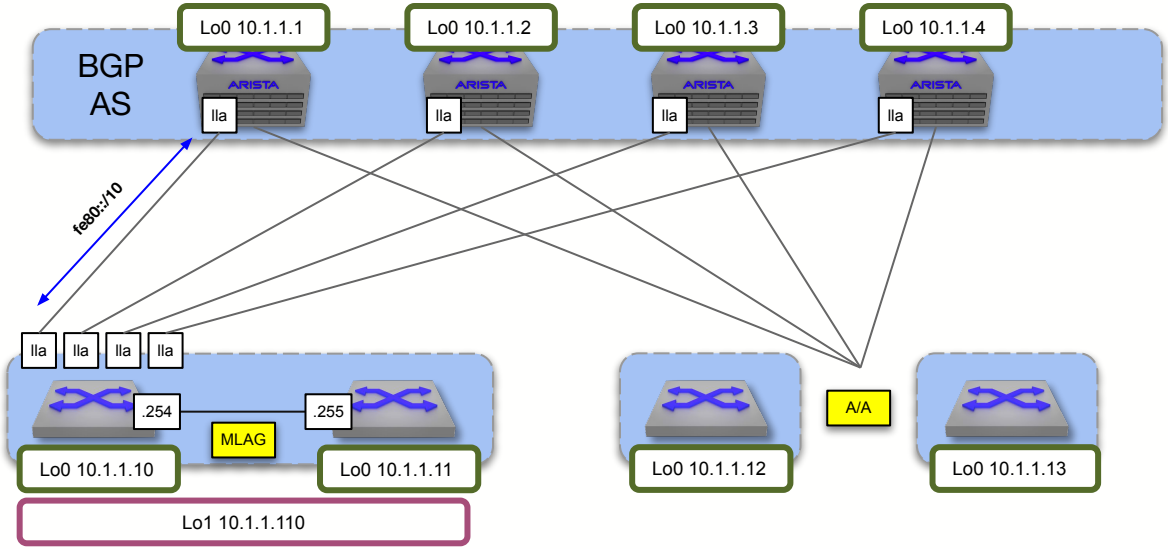
**Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop**

Internet Engineering Task Force (IETF)
Request for Comments: 8950
Obsoletes: 5549
Category: Standards Track
Published: November 2020
ISSN: 2070-1721

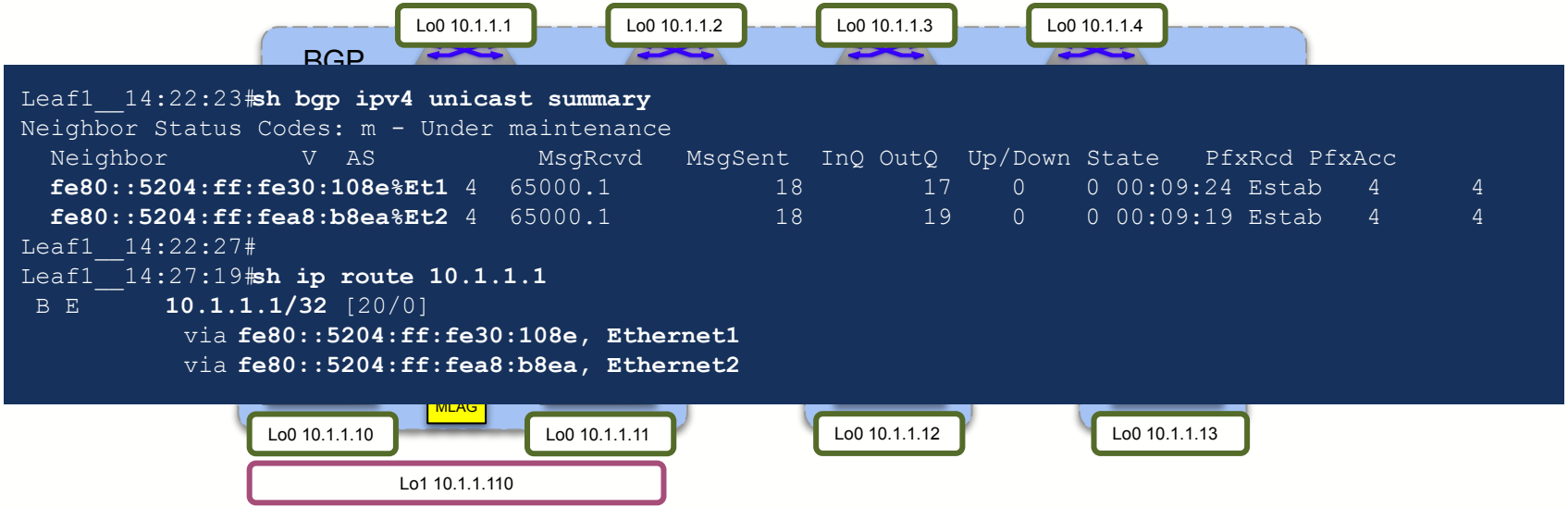
S. Litkowski
Cisco
S. Agrawal
Cisco
K. Ananthamurthy
Cisco
K. Patel
Arrcus

**Advertising IPv4 Network Layer Reachability Information (NLRI) with an IPv6 Next Hop**

# Quite a common concept … but in data centers



# Quite a common concept … but in data centers



# RFC 5549: IPv4 Unicast NLRI with IPv6 Next Hops

- Some AFIs/SAFIs in BGP allow the next-hop address to belong to a different address family

| | | |
|---|---|---|
| **MP_REACH_NLRI** | AFI | 1 |
| | SAFI | 1, 2, 4 or 128 |
| | Length of Next Hop Address | 16 or 32 bytes |
| | Next Hop Address | IPv6 address of Next Hop |
| | NLRI | NLRI as per current AFI/SAFI selection |

# RFC 5549 vs. 8950

**RFC 8950 is extending the NLRI behavior of RFC 5549**

- Multicast VPN Support added
- NLRI encoding change for AFI/SAFI 1/128
  - Bringing consistency to next hop encoding for ‘VPNv4 over IPv4’ and ‘VPNv6 over IPv6’
  - Not backwards compatible/interoperable for AFI/SAFI 1/128 of RFC 5549 due to NH field change

- RFC 8950 is mostly backwards interoperable with RFC 5549
  - No impact for ‘standard’ BGP IPv4 Unicast use case
  - Exception: VPNv4 over IPv6-only Cores (RFC 8950 Section 6.2)

# RFC 8950: RFC 5549 with Extended VPN support

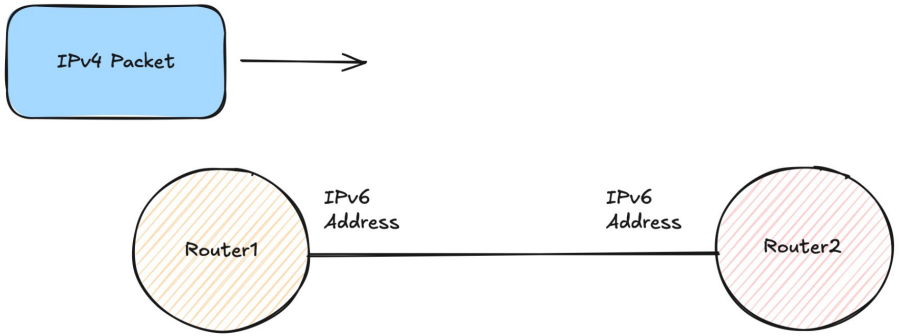
| | | |
|---|---|---|
| **MP_REACH_NLRI** | AFI | 1 |
| | SAFI | 1, 2, or 4 |
| | Length of Next Hop Address | 16 or 32 bytes |
| | Next Hop Address | IPv6 address of Next Hop |
| | NLRI | NLRI as per current AFI/SAFI selection |

| | | |
|---|---|---|
| **MP_REACH_NLRI** | AFI | 1 |
| | SAFI | 128 or 129 |
| | Length of Next Hop Address | 24 or 48 bytes |
| | Next Hop Address | VPN IPv6 address of a NH with an 8-octet RD |
| | NLRI | NLRI as per current AFI/SAFI selection |

# How does it work?

- R1 receives an IPv4 prefix via an IPv4 Unicast BGP session from an IPv6 neighbor
- R1 receives an IPv4 packet and wants to forward it
  - R1 looks up the destination for the IPv4 prefix and finds an IPv6 next-hop
  - R1 looks up the MAC for the IPv6 next-hop via IPv6 neighbor discovery
- R1 forwards packet to outgoing interface for MAC address of R2

**No tunneling (encap/decap) involved**



# Is this a supported feature?

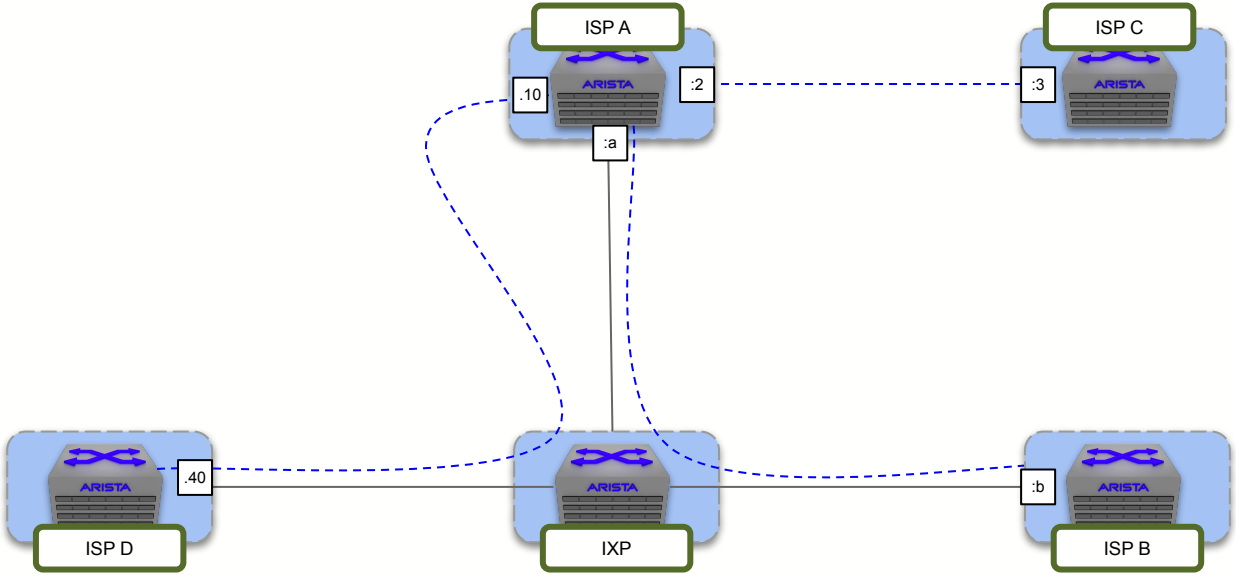
An independent collection of platforms supporting this feature created by the **RFC 8950 Working Group** of Euro-IX

Configuration examples are also available in the GitHub repository

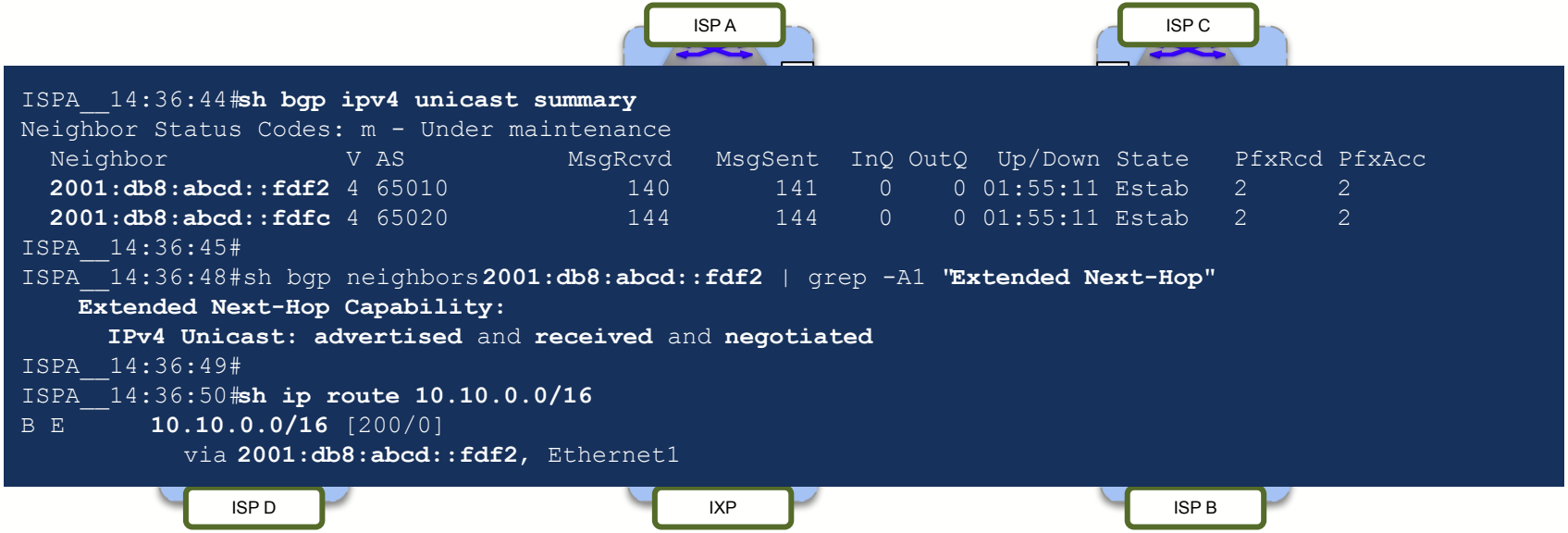
| Vendor | Platform | Software Version | Notes |
|---|---|---|---|
| Arista | EOS | 4.22.1F | Some support already in 4.17 |
| Cisco | IOS XE | not supported | IPv6 next hop support for VPN routes since 17.8.1 |
| Cisco | IOS XR | 7.3.3 | |
| Cisco | NX-OS | ? | to be tested |
| CZNIC | Bird | 2.0.8 | RIB-only since 2.0.0; Linux kernel version 5.2 required |
| Exa | ExaBGP | 4.1.0 | Cannot program Linux netlinks for RFC5549 |
| Extreme Networks | IronWare, SLX-OS | not supported | verified with vendor |
| Juniper | Junos | 17.3R1 | |
| Mikrotik | ROS | not supported | |
| NetDEF | FRR | 7.0.0 | Linux kernel version 5.2 required |
| Nokia | SR-OS | 20.2.R1 | |
| Nokia | SR Linux | 20.06 | to be tested |
| OSRG | GoBGP | supported for several years | no FIB integration tested |
| RSSF | OpenBGPd | not supported | on the roadmap |
| Edgecore | OCNOS | not supported in 1.3.8 | Awaiting further comment from OCNOS developers |
| Vyatta | VyOS | 1.2.2 | See FRR above |

**https://github.com/euro-ix/rfc8950-ixp**

# Adoption of Service Providers



# Adoption of Service Providers



# Peering Edge

# Security best practices

# Peering Edge Security Best Practices

| | | |
|---|---|---|
| ACL | CoPP | Max Prefix Limit |
| Prefix Filtering<br>(Prefix/AS Path Lists) | MD5 Authentication | TTL Security Check (GTSM) |
| TCP-AO Authentication | RPKI | Logging |

# Peering Edge Security Best Practices

| | | |
|---|---|---|
| ACL<br>OLD | CoPP<br>DONE ALREADY | Max Prefix Limit<br>GET CREATIVE |
| Prefix Filtering (Prefix AS Path Lists)<br>BINGO | MD5 Authentication<br>ANCIENT | **TTL Security Check (GTSM)** |
| **TCP-AO** | RPKI<br>AGAIN? | Logging<br>BORING |

# Potential Attack Vectors

**DoS attacks**

**SYN flooding**

**TCP FIN/RST attacks**

**TCP Session hijacking**

**Man in the Middle**

**Replay Attacks**

**Route hijacking attempts**

# Securing TCP Sessions


**Generalized TTL Security Mechanism (GTSM)**

**TCP MD5**

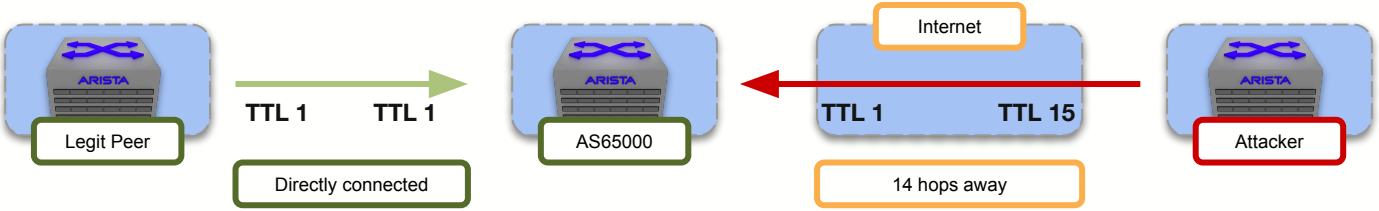
**TLS**

**IPsec**

**SYN Cookies**

**RFC 4953: Defending TCP Against Spoofing Attacks**

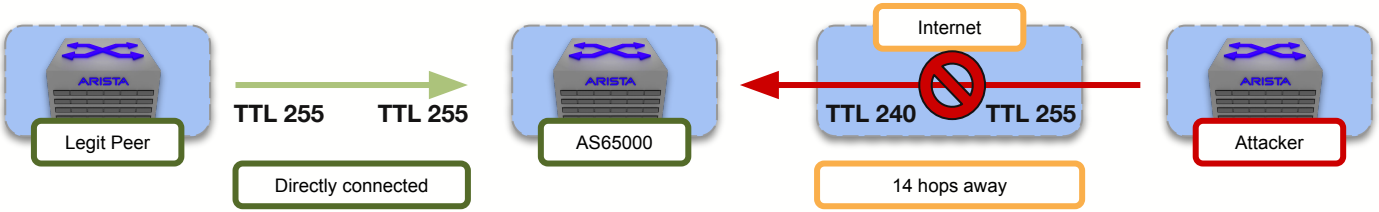
# DoS Attack Vector




- TTL/Hop limit = 1 is default for eBGP sessions
  - Most of the time we do not want multi-hop sessions anyways
- Remote attacker could adjust TTL and spoof packets
  - Can cause TCP session resets and increase CPU utilization (starvation attack)

# Generalized TTL Security Mechanism

- Uses TTL (IPv4) or Hop Limit (IPv6) attributes to protect
- Enable GTSM (RFC 3682) for **directly connected eBGP sessions**
  - **Packets will be transmitted with TTL 255 (or configured value)**
  - **Packets received with TTL 255 are discarded**
- Useful for peers with Public IP PtP networks configured
  - IXP networks are usually not routed globally
  - Customer connections/PNIs usually are routed



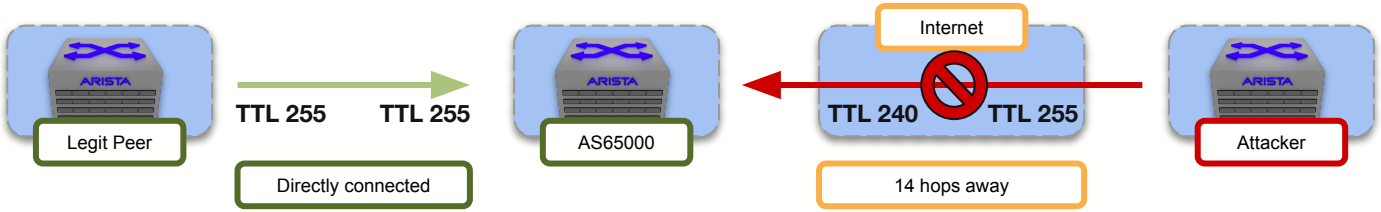
# Generalized TTL Security Mechanism

- Uses TTL (IPv4) or Hop Limit (IPv6) attributes to protect

```
AS65000# show bgp neighbors 2001:db8:abcd::fdf2 | i TTL
TTL is 1

AS65000(config)# router bgp 65000
AS65000(config-router-bgp)#neighbor 2001:db8:abcd::fdf2 ttl maximum-hops 1

AS65000# show bgp neighbors 2001:db8:abcd::fdf2 | i TTL
TTL is 255, BGP neighbor may be up to 1 hops away
```



# A normal BGP configuration …

```
router bgp 65000
   router-id 10.255.255.42
   bgp missing-policy direction in action deny
   bgp missing-policy direction out action deny
   neighbor ISP peer group
   neighbor ISP ttl maximum-hops 1
   neighbor ISP route-map EVERYTHING in
   neighbor ISP route-map OWN-AS out
   neighbor 10.11.12.5 peer group ISP
   neighbor 10.11.12.5 remote-as 65001
   neighbor 10.11.12.5 password 7 42yEZ7Db8KU/4m8Is9OcJw==
```

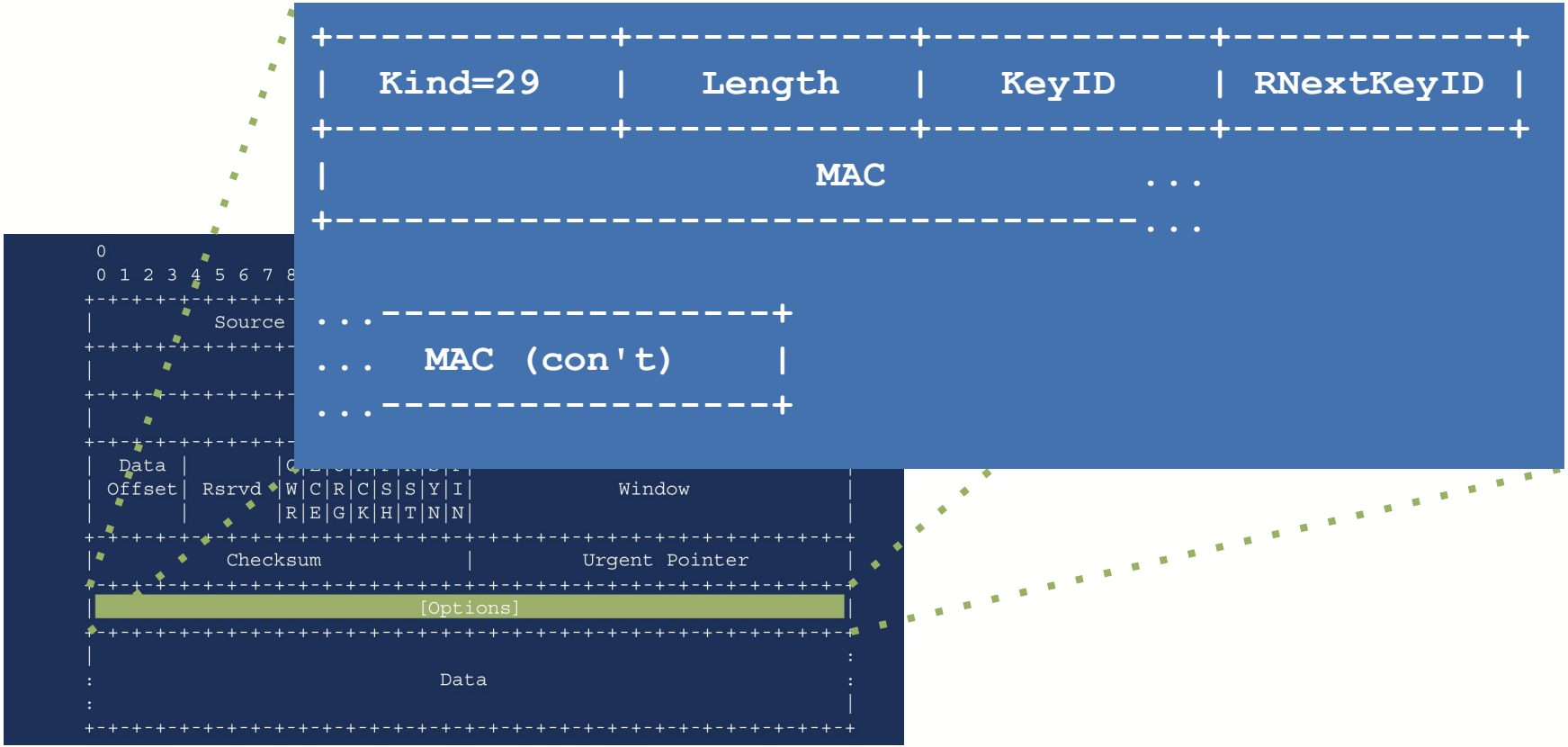
# … with a problem!

```
router bgp 65000
   router-id 10.255.255.42
   bgp missing-policy direction in action deny
   bgp missing-policy direction out action deny
   neighbor ISP peer group
   neighbor ISP ttl maximum-hops 1
   neighbor ISP route-map EVERYTHING in
   neighbor ISP route-map OWN-AS out
   neighbor 10.11.12.5 peer group ISP
   neighbor 10.11.12.5 remote-as 65001
   neighbor 10.11.12.5 password 7 42yEZ7Db8KU/4m8Is9OcJw==
```

# TCP Authentication Option (RFC 5925)

- Allows the user to authenticate TCP segments
  - Huge improvement over TCP MD5
- Provides stronger hashing algorithms
- Offers protection against replay attacks
- Has better key management
- Can rotate keys without resetting the TCP connection

# TCP-AO is part of the TCP Header



# TCP-AO Concepts

| **Master Key Tuple** | Traffic Key | Message Authentication Code (MAC) |
|---|---|---|

- TCP connection identifier
- TCP option flag
- IDs (KeyID / RNextKeyID)
- Master Key
- Key Derivation Function (KDF)
- Message Authentication Code (MAC) algorithm

- Properties of MKTs
  - MKT parameters are not changed.
  - New MKTs can be installed
  - Connection can change which MKT it uses

# TCP-AO Concepts

| Master Key Tuple | **Traffic Key** | Message Authentication Code (MAC) |
|---|---|---|

- MKT (master key)
- Local and remote IP address pairs
- TCP port numbers
- TCP Initial Sequence Numbers (ISNs) in each direction

- Keys are unidirectional!

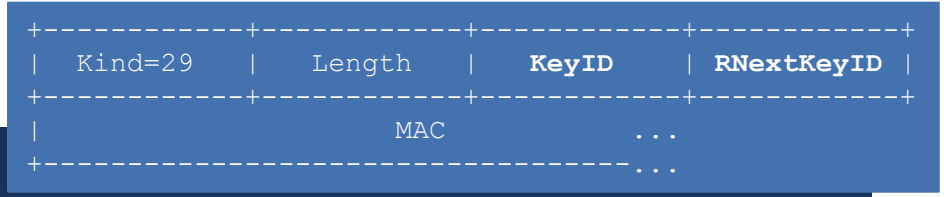
- Mandatory algorithms (RFC 5926):
  - KDF_HMAC_SHA1
  - KDF_AES_128_CMAC

# TCP-AO Concepts

| Master Key Tuple | Traffic Key | **Message Authentication Code (MAC)** |
|---|---|---|

- Mandatory algorithms (RFC 5926):
  - HMAC-SHA-1-96
  - AES-128-CMAC-96
- Calculating of MAC based on:
  - Sequence Number Extension (SNE) → Replay protection!
  - IP Pseudo Header (as used for the TCP checksum)
  - TCP header
  - TCP data

# How would configuration look like?




```
management security
   session shared-secret profile BGP
      secret 10 7 $1c$zXHy2/5IOz6JEc5qRNYMBA== receive-lifetime 2023-01-01 00:00:00 infinite \
         transmit-lifetime 2023-01-01 00:00:00 infinite
      secret 0 7 $1c$zXHy2/5IOz6JEc5qRNYMBA== receive-lifetime 2023-01-01 00:00:00 infinite \
         transmit-lifetime 2023-01-01 00:00:00 infinite

router bgp 65006
   neighbor 10.255.255.5 remote-as 65001
   neighbor 10.255.255.5 password shared-secret profile BGP algorithm hmac-sha1-96
   neighbor 10.255.255.10 remote-as 65002
   neighbor 10.255.255.10 password shared-secret profile BGP algorithm aes-128-cmac-96
```

**https://www.arista.com/en/support/toi/eos-4-28-2f/16087-bgp-tcp-authentication-option-tcp-ao**

# TCP-AO Configuration Examples

**https://github.com/TCP-AO/Configuration-examples**