BGP Session Security Mechanisms

This chapter provides an overview of essential security mechanisms for protecting BGP sessions on Cisco routers. It covers key features such as BGP keychains, Martian address checks, TTL security (GTSM), interface-based LPTS identifiers, and prefix origin validation using RPKI. Each section explains the purpose of the mechanism and offers practical configuration guidance to help secure BGP routing against common threats.

BGP keychains

A BGP keychain is a security mechanism that

- enables keychain authentication between two BGP peers based on standardized protocols
- allows hitless key rollover for authentication using time-based specifications, and
- provides a configurable tolerance window to handle clock skew between endpoints for seamless operation.

Keychain interoperability and behavior

Both BGP endpoints must comply with the draft-bonica-tcp-auth-05.txt standard for keychain authentication to function. A keychain on one endpoint and a password on the other will not work. The configurable tolerance window extends the accept period to allow for clock differences and maintains hitless key rollover for applications such as routing and management protocols.

If there is a keychain configuration mismatch at the endpoints resulting in no common keys, BGP session traffic (send or accept) may be interrupted. Otherwise, the key rollover does not disrupt the BGP session.

Configure keychains for BGP

Configure BGP keychains to secure authentication for BGP sessions using MAC authentication algorithms and enable graceful key rollover.

BGP keychains enhance the security of BGP routing by providing flexible authentication options and key management. This is especially useful in environments where multiple neighbors or session groups need secure, easily managed authentication.

Before you begin

- Ensure you have a defined keychain with the necessary keys and authentication parameters.
- Identify the autonomous system (AS) numbers for your router and remote neighbors.

Procedure

Enter BGP configuration mode, and configure keychain-based authentication for the neighbor.

Example:

```
Router# configure
Router(config)# router bgp 120
Router(config-bgp)# neighbor 172.16.40.24
Router(config-bgp-nbr)# remote-as 2002
Router(config-bgp-nbr)# keychain kych_a
```

Note
If a keychain is configured for a neighbor group or session group, a neighbor using the group inherits the keychain. Values configured directly for a neighbor override any inherited values.

Martian address checks

A Martian address check is a router security feature that

- prevents routers from accepting packets with reserved or illogical IP address prefixes
- is applied by default in BGP configurations to drop packets originating from Martian addresses, and
- can be disabled to allow routers to process routes from specific sites using designated IPv4 or IPv6 prefixes.

Martian addresses are reserved or undefined IP address ranges that should not appear in legitimate internet routing tables. Filtering these addresses improves network security by helping ensure that only valid, routable addresses are accepted during routing.

Examples

Common Martian address prefixes include:

- IPv4:

- 0.0.0.0/8
- 127.0.0.0/8
- 224.0.0.0/4

- IPv6:
  - ::
  - ::0002 through ::ffff
  - ::ffff:a.b.c.d
  - fe80:xxxx
  - ffxx:xxxx

Restrictions:

Routers running OSPF or IS-IS protocols cannot access routes with Martian address prefixes, even if the Martian address check is disabled.

Disable the Martian address check in BGP

By default, Cisco routers drop routes and packets with Martian (reserved or unusual) IP prefixes during BGP operations. You may need to override this security check to allow routing for certain special network scenarios.

Before you begin

Make sure you have console or privileged EXEC access to the Cisco 8000 Series Router.

Procedure

Step 1 Enter router BGP configuration mode, and use the **default-martian-check disable** command to disable the Martian address check.

Example:

```
Router# configure
Router(config)# router bgp 100
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# default-martian-check disable
Router(config-bgp-af)# commit
```

Step 2 Use the **show bgp ipv4 unicast** command or **show bgp ipv6 unicast** command to check whether the Martian address check is enabled or disabled in your BGP configuration.

Example:

```
Router# show bgp ipv6 unicast
BGP router identifier 10.2.2.1, local AS number 1
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0800000 RD version: 29
BGP main routing table version 29
```

```
BGP NSR Initial initsync version 4 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
Dampening enabled
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
Network                  Next Hop           Metric   LocPrf     Weight Path
*>i::/0              1:1:1:1:1:1:1:1         100        0             i
* i192:1::/112       1.1.1.1                   0      100             0 ?
*>i                  1:1:1:1:1:1:1:1           0      100             0 ?
* iff11:1123::/64    1.1.1.1                   2      100             0 ?
*>i                  1:1:1:1:1:1:1:1           2      100             0 ?
```

BGP eBGP security GTSM

BGP eBGP security GTSM is a BGP security feature that

- restricts accepted IP packets to those with a Time to Live (TTL) or Hop Limit equal to the maximum value for eBGP neighbors
- protects a router's control plane from CPU-utilization attacks caused by forged protocol packets, and
- applies robust session security for eBGP peerings, especially between directly connected or loopback-adjacent routers.

Table 1: Feature history table

Feature Name	Release Information	Feature Description
BGP-eBGP Security GTSM	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
BGP-eBGP Security GTSM	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release Information	Feature Description
BGP-eBGP Security GTSM	Release 7.3.1	The Generalized TTL Security Mechanism (GTSM) is designed to protect a router's IP-based control plane from CPU-utilization based attacks. This feature enables the router to accept only IP packets with a TTL count that is equal to the maximum TTL value. New command introduced: • **ttl-security**

Generalized TTL Security Mechanism (GTSM)

GTSM leverages the fact that most protocol peerings occur between adjacent routers or loopback addresses. Since TTL spoofing is nearly impossible in these scenarios, enforcing maximum TTL value acceptance creates a simple and effective defense against infrastructure attacks using forged packets. GTSM applies to both IPv4 (TTL) and IPv6 (Hop Limit) sessions.

How GTSM Works

When GTSM is enabled, the router only accepts packets whose TTL equals the maximum value. Packets with lower TTL are discarded and do not generate ICMP responses, preventing feedback to attackers.

Configure BGP eBGP security GTSM

Secure eBGP neighbor sessions using the Generalized TTL Security Mechanism (GTSM).

Before you begin

Identify the eBGP neighbor address and relevant autonomous system numbers.

Procedure

Step 1 Enter router BGP configuration mode, set the eBGP multihop value, and use the **ttl-security** command to enable GTSM for the eBGP neighbor.

Example:

```
Router(config)# router bgp 100
Router(config-bgp)# neighbor 2001::db8
Router(config-bgp-nbr)# remote-as 200
Router(config-bgp-nbr)# ebgp-multihop 255
Router(config-bgp-nbr)# ttl-security
Router(config-bgp-nbr)# address-family ipv6 unicast
Router(config-bgp-nbr-af)# multipath
Router(config-bgp-nbr-af)# route-policy PASS_ALL in
Router(config-bgp-nbr-af)# route-policy PASS_ALL out
```

Step 2 (Optional) Enable multipath for redundancy or load balancing, and apply route policies as required.

Example:

```
Router(config)# router bgp 100
Router(config-bgp)# neighbor 2001::db8
```

```
Router(config-bgp-nbr)# address-family ipv6 unicast
Router(config-bgp-nbr-af)# multipath
Router(config-bgp-nbr-af)# route-policy PASS_ALL in
Router(config-bgp-nbr-af)# route-policy PASS_ALL out
```

Interface-based LPTS identifiers

An interface-based LPTS identifier is a network security feature that

- associates each directly connected external BGP (eBGP) neighbor with a specific router interface
- restricts inbound traffic so only packets originating from a designated eBGP neighbor can traverse through the mapped interface, and
- prevents IP spoofing and session hijacking attempts by enforcing strict interface-level packet filtering and policing.

Table 2: Feature History Table

Feature Name	Release Name	Description
Protection of Directly Connected EBGP Neighbors through Interface-Based LPTS Identifier	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8712-MOD-M • 8011-4G24Y4H-I
Protection of Directly Connected EBGP Neighbors through Interface-Based LPTS Identifier	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release Name	Description
Protection of Directly Connected EBGP Neighbors through Interface-Based LPTS Identifier	Release 7.10.1	We have enhanced the network security for directly connected eBGP neighbors by ensuring that only packets originating from designated eBGP neighbors can traverse through a single interface, thus preventing IP spoofing. This is made possible because we've now added an interface identifier for Local Packet Transport Services (LPTS). LPTS filters and polices the packets based on the type of flow rate you configure. The feature introduces these changes: **CLI:** • **bgp lpts-secure-binding** **YANG Data Model:** • `Cisco-IOS-XR-um-router-bgp-cfg` (see GitHub, YANG Data Models Navigator)

Overview of Local Packet Transport Services (LPTS) in BGP

LPTS maintains tables describing all packet flows destined for the secure domain router (SDR), ensuring packets are delivered only to their intended destinations. In BGP sessions, LPTS entries are categorized as follows:

- **BGP known:** Entries for established BGP neighbors.
- **BGP configured peer:** Entries for initial packets (TCP SYN and 3rd ACK) from specifically configured BGP neighbors.
- **BGP default entries:** Entries for all packets from unconfigured BGP neighbors.

Security enhancement with interface identifier

By adding an interface identifier to LPTS entries for directly connected eBGP neighbors, the router ensures that only traffic from the designated interface and neighbor IP can match the LPTS entry and reach the BGP session. Spoofed packets from other interfaces, even with correct IP/port/VRF combinations, only match the default LPTS entry where they are policed and forwarded to TCP for reset generation. This prevents attackers from exploiting established session entries by flooding from other interfaces.

Conditions for passing the interface identifier

The interface identifier is passed to LPTS and TCP only when all these conditions are met:

- The BGP peer is configured as external (eBGP).

- Fast External Failover (FEF) is not disabled.
- The BGP peer is directly connected.
- The BGP peer is not a dynamic peer.
- eBGP multihop is not enabled.
- Default eBGP TTL is used.
- The "ignore connected" option is not configured.
- A non-link local IPv6 neighbor address is configured.

Interface identifier binding during session establishment

During session establishment, both passive (received connections) and active (initiated connections) BGP bindings supply the interface identifier, so the LPTS entry for the connection is tightly bound to the specified interface.

Interface-based LPTS identification example

Suppose an attacker floods packets matching the established BGP session (source IP, destination IP, source port, destination port, VRF) from an unintended interface. With interface-based LPTS identification enabled, those packets do not match the LPTS entry for the legitimate peer; they are discarded or strictly policed, ensuring BGP session stability and preventing flapping.

Configure LPTS secure binding for directly connected EBGP neighbors

Enable secure binding between LPTS and directly connected eBGP neighbors to enhance network protection.

Procedure

Step 1 Enter BGP configuration mode, and enable LPTS secure binding for BGP.

Example:

```
Router#(config)router bgp 100
Router#(config-bgp) bgp lpts-secure-binding
```

Step 2 Confirm that LPTS secure binding is enabled.

Example:

```
Router# show bgp process | in LPTS

Wed Dec 14 14:28:33.779 PST
LPTS secure binding is enabled
```

Step 3 Verify that LPTS entries now include interface handle identifiers.

Example:

```
Router# show lpts pifib entry brief
```

```
IPv4    default  TCP    any          [0x00000003]      10.10.10.1,23756 10.10.10.2,179
IPv4    default  TCP    any          0/0/CPU0           10.10.10.1,179 10.10.10.2
IPv4    default  TCP    Gi0/2/0/1    [0x00000003]       192.0.2.1,57342 192.0.2.3,179
IPv4    default  TCP    Gi0/2/0/1    0/0/CPU0           192.0.2.1,179 192.0.2.3
IPv4    default  TCP    any          [0x00000003]       209.165.201.1,179 209.165.201.4,52798
IPv4    default  TCP    any          0/0/CPU0           209.165.201.1,179 209.165.201.0/24
IPv4    default  TCP    Gi0/2/0/3    [0x00000003]       172.16.0.1,179 172.16.0.5,49505
IPv4    default  TCP    Gi0/2/0/3    0/0/CPU0           172.16.0.1,179 172.16.0.5
IPv4    default  TCP    any          [0x00000003]       192.168.0.1,179 192.168.0.6,32909
IPv4    default  TCP    any          0/0/CPU0           192.168.0.1,179 192.168.0.6
```

Step 4 Verify that the status of the connected interface handle in LPTS is active for the eBGP neighbor.

Example:

```
Router# show bgp neighbor 192.0.2.3, detail | in Connected

Wed Dec 14 14:28:51.814 PST
  Connected IFH: 0x1000080, IFH in LPTS 0x1000080
```

BGP prefix origin validation mechanisms

A BGP prefix origin validation mechanism is a route security feature that

- uses the Resource Public Key Infrastructure (RPKI) to validate the Autonomous System (AS) originating a BGP prefix
- prevents prefix mis-announcement by verifying that the origin AS claiming an address prefix is authorized, and
- enhances routing security by ensuring that BGP routers accept only prefixes with verifiable, legitimate origin AS numbers.

Table 3: Feature history table

Feature Name	Release Information	Feature Description
BGP Prefix Origin Validation Based on RPKI	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Feature Name	Release Information	Feature Description
BGP Prefix Origin Validation Based on RPKI	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only); Modular Systems (8800 [LC ASIC: P100])(select variants only*) This feature enhances BGP route security by using Resource Public Key Infrastructure (RPKI) to validate the origin Autonomous System (AS). It associates a route's address prefix with AS numbers, starting with the origin AS, and uses RPKI to verify the AS claiming the prefix. This helps prevent prefix mis-announcement, ensuring routes are secure and legitimate. *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Overview of RPKI

RPKI is a globally distributed cryptographic framework that creates a verifiable database of IP address blocks and Autonomous System (AS) numbers. RPKI enables network operators to securely certify which AS is authorized to advertise specific IP address prefixes.

BGP origin validation process

When a BGP router receives a route, it examines the AS_PATH attribute, which lists the sequence of ASes across which a prefix announcement has traveled. The router identifies the origin AS for the prefix and checks this against the authorization records in the RPKI database. If RPKI lists the prefix-origin pair as valid, the BGP router accepts the route. If not, the router considers the route invalid or suspicious. This process helps you prevent unauthorized or accidental route advertisements.

Security benefits

By leveraging RPKI-based BGP prefix origin validation, networks can defend against several well-known routing threats, such as prefix hijacking, mis-announcements, and monkey-in-the-middle attacks. Only routes authenticated through RPKI are trusted, reducing the risk of disrupted or maliciously re-routed Internet traffic.

Example of BGP prefix origin validation using RPKI

Suppose AS 64500 originates the prefix 192.0.2.0/24 and announces it into BGP. Another router receives this route and checks the RPKI database. If the database confirms that AS 64500 is authorized to originate 192.0.2.0/24, the route is considered valid. If not, the route is rejected or marked as suspicious, thus preventing a possible hijack or mis-announcement.

Configure an RPKI cache server

RPKI helps prevent route hijacking by verifying that BGP routes are correctly originated. Configuring a cache server allows the router to obtain validated prefix information for secure routing decisions.

Before you begin

- Obtain the RPKI cache server's IP address or hostname and transport requirements (SSH or TCP).
- Have SSH credentials available if using SSH as the transport protocol.

Procedure

Step 1 Enter RPKI cache server configuration mode and configure the transport protocol (TCP or SSH) and port for the cache server.

Example:

```
Router(config)# router bgp 100
Router(config-bgp)# rpki server 10.2.3.4
Router(config-bgp-rpki-server)# transport ssh port 22
```

Note
The default SSH port is 22. Both SSH and TCP support ports in the range 1–65535.

Tip
You can set the transport to either TCP or SSH. Changing the transport method causes the cache session to flap.

Step 2 (Optional, when using SSH) Set the username and password for the cache server.

Example:

```
Router(config-bgp-rpki-server)# username ssh_rpki_cache
Router(config-bgp-rpki-server)# password ssh_rpki_pass
```

Step 3 (Optional) Configure the preference for this cache server if multiple servers are used.

Example:

```
Router(config-bgp-rpki-server)# preference 1
```

Range for the preference value is 1 to 10. Lower values have higher priority.

Step 4 (Optional) Set the purge time for how long BGP retains route information after the cache session drops.

Example:

```
Router(config-bgp-rpki-server)# purge-time 30
```

Range for the purge time is 30 to 360 seconds.

Step 5 (Optional) Configure periodic refresh and response timers.

To set the refresh interval or disable it:

```
Router(config-bgp-rpki-server)# refresh-time 20
```

Or

```
Router(config-bgp-rpki-server)# refresh-time off
```

To set the maximum response wait time or disable the timeout:

```
Router(config-bgp-rpki-server)# response-time 30
```

Or

```
Router(config-bgp-rpki-server)# response-time off
```

Step 6 (Optional) Shut down the RPKI cache server.

Example:

```
Router(config-bgp-rpki-server)# shutdown
```