



An Introduction to BGP Hijacking & Leaking



BGP Hijacking & Leaking Defined

- + BGP Hijacking (aka “BGP Leaking”) is when a BGP peer does something which causes BGP traffic to be re-routed over the Internet.
- + This can be done intentionally (maliciously) or accidentally
- + *“Attackers need to control or compromise a BGP-enabled router that bridges between one autonomous system (AS) and another”*
- + Where does the re-routed traffic go in a Hijack attack?
 - + Black holed
 - + Re-routed to malicious destination servers
 - + Re-routed through malicious eavesdroppers
 - + Re-routed through non-optimal paths

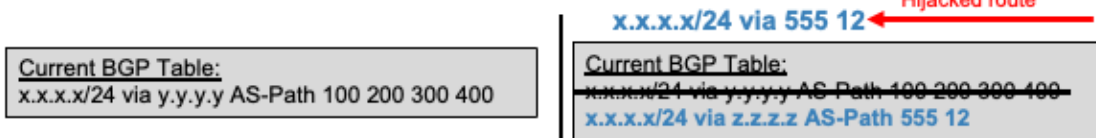


- Quote taken from <https://www.cloudflare.com/learning/security/glossary/bgp-hijacking>

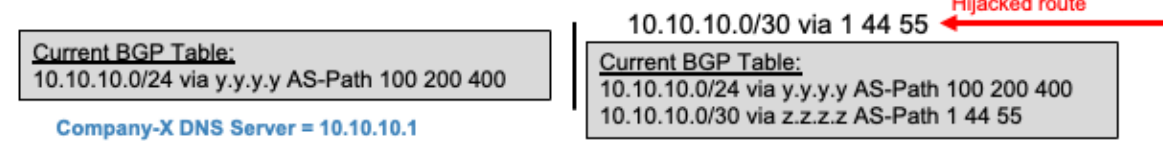
Mechanisms Used in Hijacking

- + In order to hijack a route one of two things must happen:

- + A duplicate BGP prefix is received with a better path



- + A more specific route is advertised to a destination



- + This can only be originated from a current BGP peer

Methods of Securing BGP

- + There are many methods of securing BGP which fall into three categories
 - + Securing BGP session establishment (protecting from rogue BGP peers)
 - + BGP Authentication
 - + Protection against receiving BGP updates for undesirable prefixes
 - + BGP Prefix Filtering
 - + BGP AS_Path Filtering
 - + Validating the origin of received BGP prefixes
 - + Remove eBGP-Multihop
 - + BGP Route Origin Authentication



- When eBGP-Multihop is configured, a BGP peer can send you a prefix with any spoofed next-hop and your router will accept it as long as you have a route to that next-hop.





BGP Authentication



BGP Authentication Overview

- + Two methods exist to authenticate BGP peers and sessions
 - + MD5 authentication
 - + TCP authentication
- + MD5 authentication
 - + Easy to configure (single command)
 - + Simple MD5 hashing of per peer passwords
 - + Passwords are static and unchanging
- + TCP authentication
 - + More complex to configure (requires key chain)
 - + Advanced cryptography used against peer passwords
 - + Passwords can rotate on a configured time basis



BGP MD5 Authentication

- + To configure MD5 authentication for BGP, use the following command:
 - + *neighbor [neighbor-ip] password <key>* (BGP subcommand)
- + This command must be configured on both routers.
- + If keys do not match or this command is only configured on one router, peer-establishment will not be formed.



BGP MD5 Authentication Example

No.	Time	Source	Destination	Protocol	Length	Info
-	3 0.698819	1.1.1.1	1.1.1.2	TCP	78	43938 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 MD5
+ Frame 3: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface -, id 0						
+ Ethernet II, Src: 0c:81:3d:1c:00:00 (0c:81:3d:1c:00:00), Dst: 0c:70:96:ee:00:00 (0c:70:96:ee:00:00)						
+ Internet Protocol Version 4, Src: 1.1.1.1, Dst: 1.1.1.2						
+ Transmission Control Protocol, Src Port: 43938, Dst Port: 179, Seq: 0, Len: 0						
Source Port: 43938						
Destination Port: 179						

+ Output omitted for brevity...

```
- Options: (24 bytes), Maximum segment size, TCP MD5 signature, End of Option List (EOL)
+ TCP Option - Maximum segment size: 1460 bytes
- TCP Option - TCP MD5 signature
  Kind: MD5 Signature Option (19)
  Length: 18
  MD5 digest: 9f79b57bf83c6cc5d8e9988ea77e3206
```



- With MD5 authentication, the MD5 Hash Digest is created using the password that was supplied in the IOS command as well as the contents of the BGP message (including the BGP Marker field).

BGP TCP AO

- + TCP AO = TCP Authentication Option
- + Required commands:

```
key chain INE tcp
key 1
  send-id 2
  recv-id 1
  cryptographic-algorithm hmac-sha-256
  key-string 0123456789abcdef0123456789abcdef
```

```
router bgp 2
  bgp log-neighbor-changes
  neighbor 1.1.1.1 remote-as 1
  neighbor 1.1.1.1 ao INE
```

- + Optionally you can add to your keychain;
 - + accept lifetime <keyword values>
 - + send lifetime <keyword values>



- This feature is not currently available in mainline IOS. You must be using a specialized version of IOS such as IOS-XE.
- The Send-ID and Recv-ID must match on both sides. Although you won't see these values in any Sniffer output, they are used in conjunction with the key-string to create the hashed digest of the key as well as verify the hashed digest of received keys.
- In the most secure environment, you'd have separate key-chains for each BGP peer, each using a different "send" and "recv" ID. However you CAN use one key-chain and share it among all your BGP peers as long as they are using the same sets of send and recv-ids.

Example of BGP TCP Authentication

No.	Time	Source	Destination	Protocol	Length	Info
6	9.416687	1.3.3.3	1.3.3.1	TCP	78	32918 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1460
↳ Ethernet II, Src: 0c:a4:f8:c9:00:01 (0c:a4:f8:c9:00:01), Dst: 0c:92:af:73:00:01 (0c:92:af:73:00:01)						
↳ Internet Protocol Version 4, Src: 1.3.3.3, Dst: 1.3.3.1						
↳ Transmission Control Protocol, Src Port: 32918, Dst Port: 179, Seq: 0, Len: 0						
Source Port: 32918						
Destination Port: 179						

+ Output omitted for brevity...

- ↳ Options: (24 bytes), Maximum segment size, Unknown (0x1d)
 - ↳ TCP Option - Maximum segment size: 1460 bytes
 - ↳ TCP Option - Unknown
 - Kind: TCP Authentication Option (29)
 - Length: 20
 - Payload: 0201e291a610cc01e99591beedcca3e91492





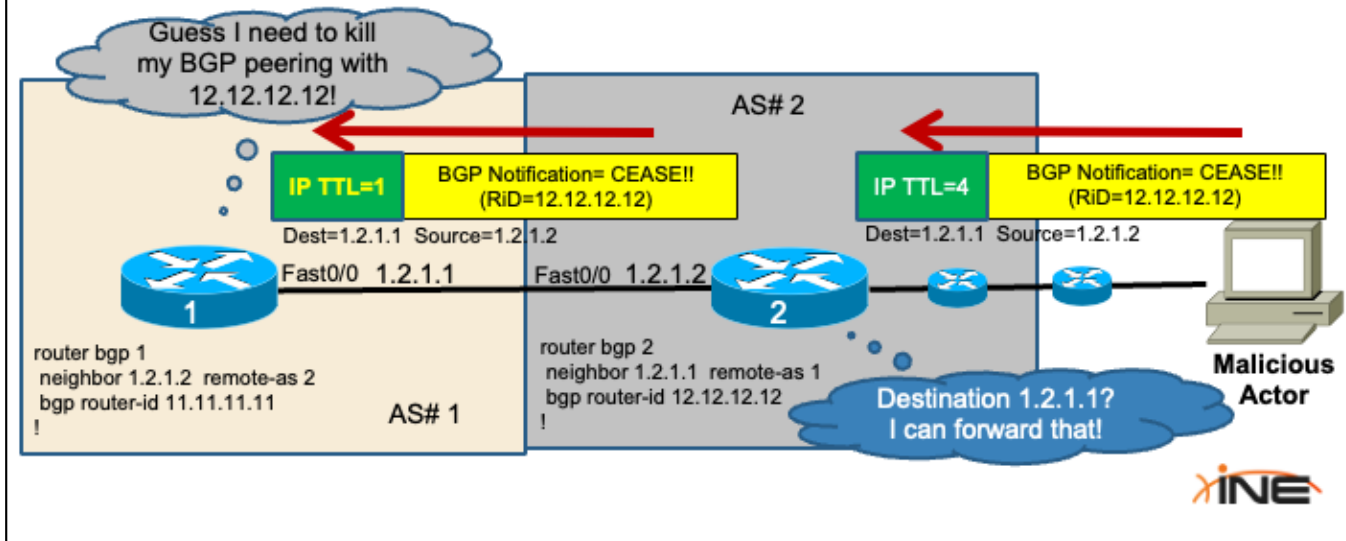


BGP TTL-Security



BGP DoS Example

eBGP's reliance on TTL=1 leaves it open to attack.



- In this case, the “Malicious Actor” has some kind of traffic generator on his PC that is capable of creating a BGP Notification message and spoofing the source IP address.
- -
- Router-1 only accepts BGP packets from Router-2 under the condition they have a TTL of “1” because Router-2 is an eBGP peer.
- -
- Ebgp-multihop wouldn't enforce any security either. It doesn't care about the TTL of incoming BGP packets from a peer as long as the TTL is one (1) or greater.

TTL and eBGP Sessions

- + eBGP sessions assume neighbor is directly-connected.
- + TTL in eBGP sessions set to "1" if Connected route is found.
- + If neighbor NOT directly connected, additional configuration needed to start BGP peering process (which affects outbound TTL)
 - + eBGP-multihop (sets TTL in outbound BGP packets to 255)
 - + Disable-connected-check (sets TTL to "1" in outbound BGP packets.
 - + TTL-Security (to be discussed next)



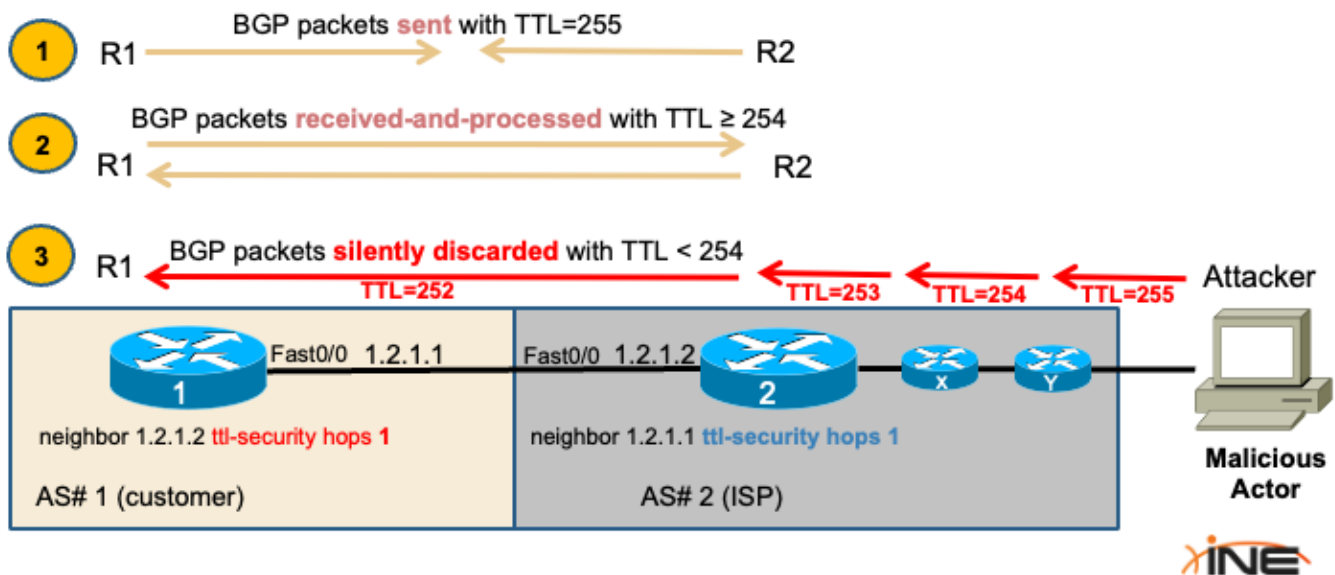
TTL-Security

- + By default, any TTL value (>0) of received BGP packets is accepted from eBGP peers.
- + TTL-Security = Mechanism to enforce TTL values to prevent DoS
 - + (config-rtr)#neighbor x.x.x.x ttl-security hops <1-254>
- + How is “hops” used?
 - + $255 - \text{hops} = X$
 - + All incoming BGP packets must have $\text{TTL} \geq X$



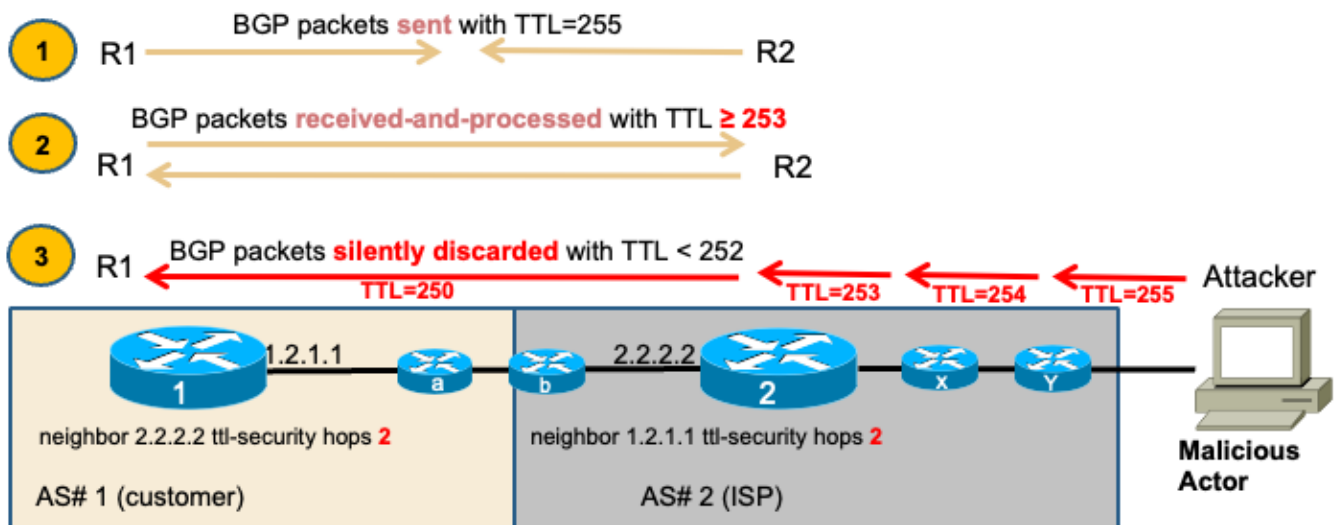
- TTL-Security is a feature that was designed to protect your router from malicious DoS attacks coming from OUTSIDE of your Autonomous System.
- -
- It is assumed that you have control over your own Autonomous-System and people from within your own company aren't trying to bring down your routers.

TTL-Security with Direct-Connection Peering



- TTL-Security (like ebgp-multihop) affects the TTL of outbound BGP packets by setting them to 255. Normally, when connected to a directly-connected eBGP peer the TTL is set to 1 in transmitted packets.
- -
- Cisco Configuration Guides state that TTL-Security should be configured on both BGP peers.
- -
- In this situation, that would be mandatory.
- -
- If this feature were only to be configured on Router-1 and not Router-2:
- ---Router-2 would see Router-1 as a directly-connected peer and generate BGP packets with the default TTL of 1.
- ---Router-1 would drop those packets from Router-2 because of TTL-Security (they don't have a TTL of 254, or 255).
- -
- If the attacker were directly-connected to Router-2 then conceivably he could thwart TTL security. However that would imply that the attacker resided within the ISP's Autonomous System (possibly as an employee) which means...you've got much bigger problems to worry about.

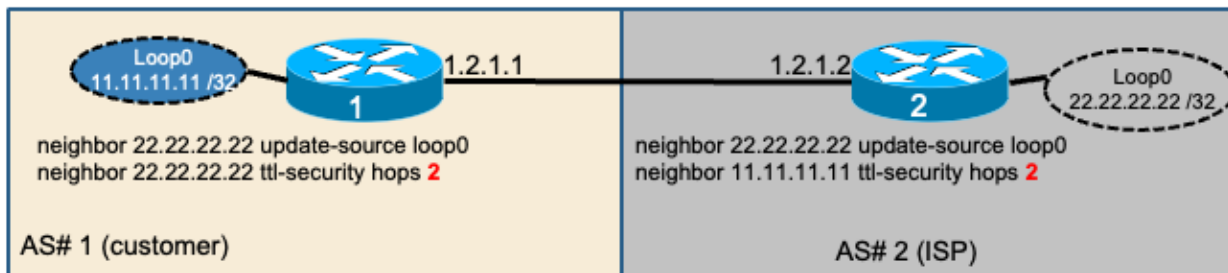
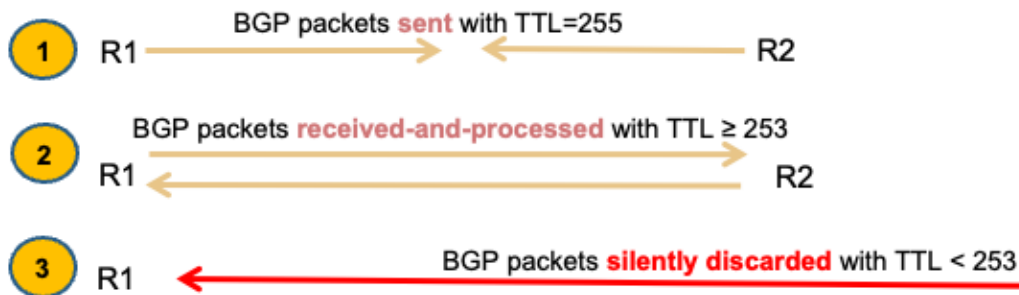
TTL-Security with Multihop Peering



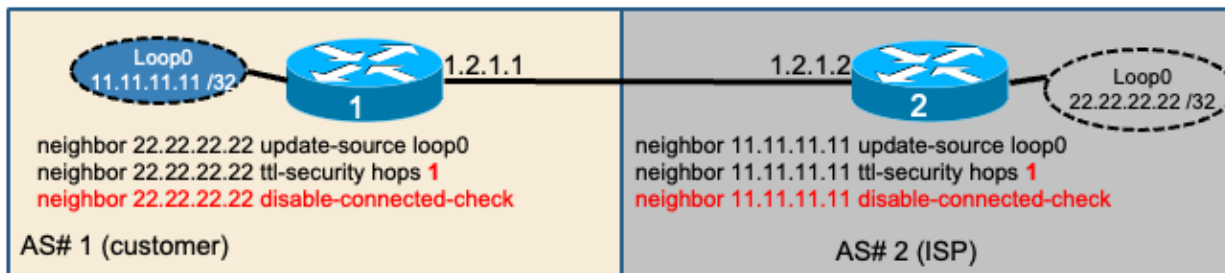
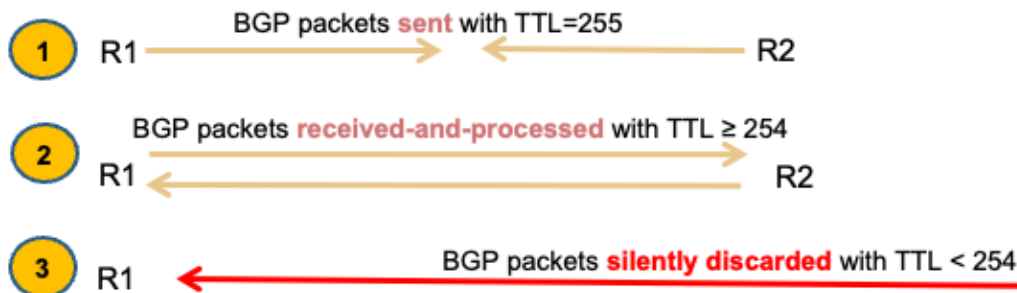
This feature cannot be used along with eBGP-Multihop feature. They are mutually exclusive.

- -

TTL-Security with Loopback Peering (Method #1)



TTL-Security with Loopback Peering (Method #2)



- Although TTL-Security is described in Cisco documentation as a security mechanism for checking received packets, we've already seen that it also influences outbound packets (by setting their TTL to 255).
- -
- If the TTL-Security feature is given a hopcount of "1" it automatically assumes the peer must be directly-connected (even though it will STILL cause egress BGP packets to be transmitted with a TTL of 255). In this case, if the IP address the router is trying to peer with (Loopback of remote router) is NOT directly-connected TTL-Security (with a hopcount of "1") will prevent the router from even starting the BGP process.
- -
- For the reason stated above, in this topology the additional command of "disable-connected-check" is needed. Recall that TTL-Security is mutually exclusive of the command "ebgp-multihop" so those two commands cannot be used together for the same peer.





BGP Maximum Prefixes



What Problem Is Solved?

- + Enterprise routers connecting via BGP to ISPs typically only receive a default route
- + When connecting to an MPLS-VPN or other BGP-based WAN service one would still expect to receive a modest quantity of prefixes
- + Some routers can't handle the entire global BGP table
 - + Currently 938,312 prefixes in Global BGP Table (as of 10/16/23)
- + A misconfigured BGP peer may accidentally send you **too many prefixes**, causing impact to your router's CPU and memory.



BGP Maximum-Prefix

- + Prior to configuring the Maximum-Prefix feature you must decide;
 - + What is an acceptable quantity of prefixes to receive?
 - + What action to take if inbound updates exceed that value?
 - + Restart BGP peering?
 - + Provide a warning?
- + **Neighbor <address | peer-group name> maximum-prefix <maximum> [threshold] restart [restart-interval] [warning-only]**

Maximum value of prefixes to receive before action is taken.

Optional threshold (percentage) of maximum value after which warning messages will be displayed.

Optional time interval (in minutes) to wait before peering is re-established.

Optional action of displaying a warning INSTEAD of killing the peering.



- By default, if you don't specify a "restart-interval" BGP will just stay down after maximum-prefixes has been triggered and will never automatically re-initialize the peer.

Verification

```
R2(config-router)#neighbor 22.7.22.7 maximum-prefix 8
R2(config-router)#
R2(config-router)#
R2(config-router)#
*Oct 16 19:19:46.631: %BGP-4-MAXPFX: Number of prefixes received from 22.7.22.7 (afi 0) reaches 7, max 8
```

```
*Oct 16 19:20:17.276: %BGP-4-MAXPFX: Number of prefixes received from 22.7.22.7 (afi 0) reaches 8, max 8
*Oct 16 19:20:17.276: %BGP-3-MAXPFXEXCEED: Number of prefixes received from 22.7.22.7 (afi 0): 9 exceeds limit
8
*Oct 16 19:20:17.276: %BGP-3-NOTIFICATION: sent to neighbor 22.7.22.7 6/1 (Maximum Number of Prefixes Reached)
7 bytes 00010100 000008
*Oct 16 19:20:17.276: %BGP-5-NBR RESET: Neighbor 22.7.22.7 reset (Peer over prefix limit)
*Oct 16 19:20:17.278: %BGP-5-ADJCHANGE: neighbor 22.7.22.7 Down Peer over prefix limit
*Oct 16 19:20:17.278: %BGP SESSION-5-ADJCHANGE: neighbor 22.7.22.7 IPv4 Unicast topology base removed from sess
```



- The default value at which a warning will be generated is when you receive 75% of the maximum-configured value.

Verification

```
R2#show ip bgp neighbor 22.7.22.7
BGP neighbor is 22.7.22.7, remote AS 7, external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle
```

```
For address family: IPv4 Unicast
  BGP table version 29, neighbor version 1/29
  Output queue size : 0
  Index 0, Advertise bit 0
  Address family not supported notification sent
  Slow-peer detection is disabled
  Slow-peer split-update-group dynamic is disabled
  Peer had exceeded the max. no. of prefixes configured.
  Maximum prefixes allowed 8
  Threshold for warning message 75%
  Reduce the no. of prefix and clear ip bgp 22.7.22.7 to restore peering
```





BGP Route Origin Validation & RPKI



What Problem Is Solved?

- + Every BGP prefix within a BGP Update carries two mandatory path attributes:
 - + Next-Hop
 - + AS-Path
- + If either of these attributes are modified by a malicious actor and accepted by a BGP router, bad things can happen.
- + The last ASN in the AS-Path attribute (read left-to-right) contains the originating ASN of the prefix
- + BGP Hijack attacks (aka BGP Leak attacks) occur when a malicious actor injects a BGP prefix into the global BGP table that they don't own by modifying the AS-Path.



Introduction to Route Origin Validation

- + BGP ROV allows a router to validate prefixes received within a BGP update.
- + Each **prefix and originating ASN** compared to a list of validated prefixes and ASNs
- + After validation, a prefix can be in one of three states indicating its trustworthiness
- + Routing and BGP update policies can be configured based on resulting validation codes.



Route Origin Validation Codes

- + **Valid:** Prefix and originating ASN matches validation list
- + **Invalid:**
 - + Exact prefix and mask found but originating ASN different than validated ASN
 - + Prefix found but exceeds the maximum length allowable in ROA record (*more on this later*)
- + **Not Found:** Prefix and originating ASN not found within validation list



- View the notes in the slide titled, “Viewing RPKI Table” for more information about the “Invalid” state.
- By default, a prefix that is marked Invalid is not advertised to any peer, will be withdrawn from the routing table if it was already advertised, and will not be flagged as a bestpath or considered as a candidate for multipath (unless a BGP bestpath command indicates otherwise).
- Unless a BGP bestpath command is configured indicating otherwise, the bestpath computation prefers Valid prefixes over Not Found prefixes, and both types of prefixes are advertised.
- By default, a prefix marked as Not Found is installed in the BGP routing table

Invalid Prefix Treatment Example

```
CSR1#show ip bgp 1.1.1.1/32
BGP routing table entry for 1.1.1.1/32, version 23
Paths: (1 available, no best path)
  Not advertised to any peer
  Refresh Epoch 2
  7
    7.1.7.7 from 7.1.7.7 (166.66.66.5)
      Origin IGP, metric 0, localpref 100, valid, external
      path 7FA1AA2D0778 RPKI State invalid
      rx pathid: 0, tx pathid: 0
      Updated on Oct 17 2023 19:22:10 UTC
```



How Are Prefixes Validated?

- + BGP Route Origin Validation utilizes RPKI
 - + Resource Public Key Infrastructure
- + Each Regional Internet Registry and many large Service Providers (such as Cloudflare) maintain RPKI repositories
- + RPKI Repository = cryptographically signed database of validated prefixes and originating ASNs
- + The RPKI repositories can be used in two ways;
 - + Owners of BGP prefixes can register their prefixes and ASNs with the RPKI repositories
 - + BGP operators can use repositories to verify the validity of received BGP prefixes
 - + This is called being a “*Relying Party*”



RPKI Terminology

+ ROA

- + Route Origin Authorization
- + Created by RIRs
- + Digitally signed with Private Key of RIR
- + Verification of Prefix/ASN ownership

+ RPKI Repository

- + Database holding full collection of ROAs
- + Downloaded by companies wishing to validate BGP prefixes

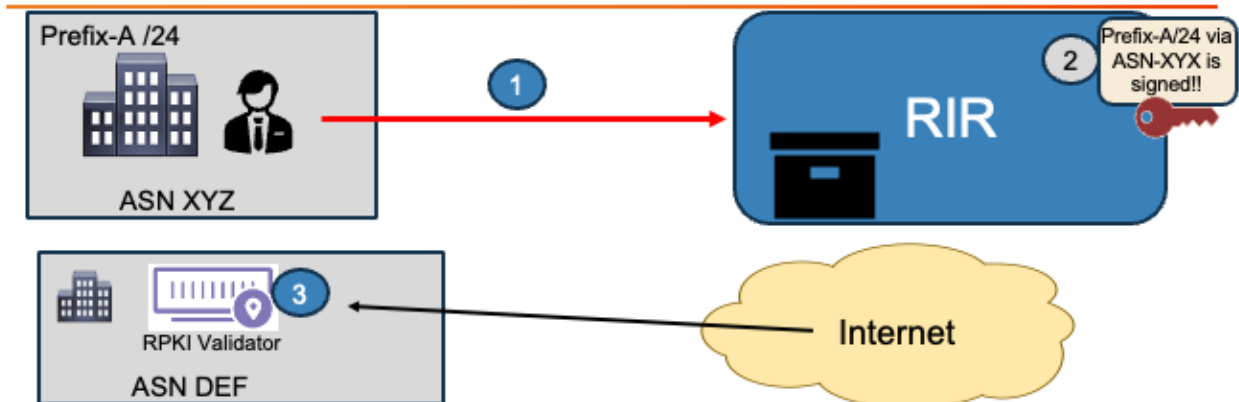
+ TAL

- + Trust Anchor Location
- + URL or other information describing the location of the RPKI Repository
- + Contains Public Key of RIR who created ROAs



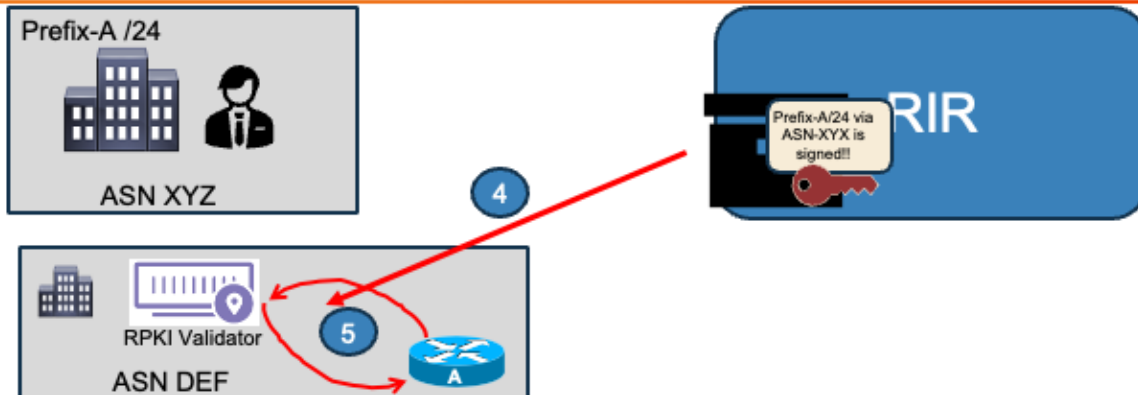
- A Trust Anchor Locator (TAL) is a file used to allow Relying Parties to retrieve RPKI data from a repository. Each Regional Internet Registry (RIR) has a TAL needed to access its RPKI repository.
- ARIN's TAL contains the URL of ARIN's published RPKI repository and ARIN's encrypted public key. The public key is used to cryptographically verify that ARIN has signed the artifacts within the repository.

Overview of RPKI Process



1. Registered authority at ASN XYZ submits signed agreements to RIR and submits ROA (Route Origin Authorization) request
2. ROA is digitally signed with Private Key of RIR and added to RIR's RPKI Database repository
3. Relying Party downloads and installs an RPKI Trust Validator application for authenticating signed ROAs

Overview of RPKI Process

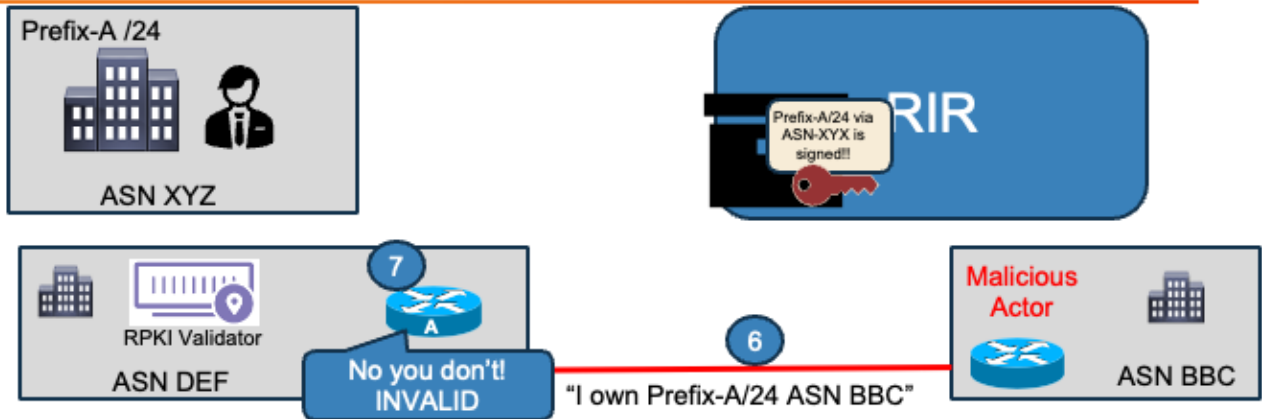


4. RPKI Validator obtains TAL (Trust Anchor Location), downloads full RPKI Repository and cache of current ROAs and authenticates each ROA
5. Local routers configured to establish TCP connection to RPKI Validator & downloads RPKI ROAs from Validator



- Because the RPKI Repo owned and managed by the RIRs frequently change, one must configure their local RPKI Validator software to periodically re-download the repository from the RIR.
- Point-4 is variable depending on the application selected to perform the RPKI Validator process. Some RPKI validators will not authenticate ROAs until a router specifically asks about it.

Overview of RPKI Process



6. Incoming BGP update received
7. Local router verifies prefix against local table of ROAs and makes a decision.



- Because the RPKI Repo owned and managed by the RIRs frequently change, one must configure their local RPKI Validator software to periodically re-download the repository from the RIR.
- Point-4 is variable depending on the application selected to perform the RPKI Validator process. Some RPKI validators will not authenticate ROAs until a router specifically asks about it.

BGP ROV Configuration

- + This command only available within some Cisco IOS versions such as IOS 15S and IOS-XE

```
router bgp 1
  bgp log-neighbor-changes
  bgp rpki server tcp 192.168.122.1 port 3323 refresh 120
  neighbor 7.1.7.7 remote-as 7
```

Your company's
RPKI Validator

RPKI Validator's
TCP port

RPKI Query
Interval



Viewing ROV Prefixes

```
CSR1#sho ip bgp
BGP table version is 16, local router ID is 7.1.7.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
Nr>	0.0.0.0	7.1.7.7				0 7 12345 i
I*	1.1.1.1/32	7.1.7.7	0			0 7 i
N*>	1.4.7.0/24	7.1.7.7	0			0 7 ?
V*>	1.7.184.0/24	7.1.7.7	0			0 7 9583 ?
Nr>	7.1.7.0/24	7.1.7.7	0			0 7 ?



Viewing Local RPKI Table

```
CSR1#sho bgp rpki table
% Command accepted but obsolete, unreleased or unsupported; see documentation.
348276 BGP sovc network entries using 55724160 bytes of memory
386489 BGP sovc record entries using 12367648 bytes of memory

Network          Maxlen  Origin-AS  Source  Neighbor
1.0.0.0/24        24      13335      0       192.168.122.1/3323
1.0.4.0/24        24      38803      0       192.168.122.1/3323
1.0.4.0/22        22      38803      0       192.168.122.1/3323
1.0.5.0/24        24      38803      0       192.168.122.1/3323
1.0.6.0/24        24      38803      0       192.168.122.1/3323
1.0.7.0/24        24      38803      0       192.168.122.1/3323
1.0.64.0/18       18      18144      0       192.168.122.1/3323
1.1.1.0/24        24      13335      0       192.168.122.1/3323
1.1.4.0/22        22      4134       0       192.168.122.1/3323
1.1.16.0/20       20      4134       0       192.168.122.1/3323
1.2.9.0/24        24      4134       0       192.168.122.1/3323
1.2.10.0/24       24      4134       0       192.168.122.1/3323
1.2.11.0/24       24      4134       0       192.168.122.1/3323
1.2.12.0/22       22      4134       0       192.168.122.1/3323
1.3.0.0/16        16      4134       0       192.168.122.1/3323
1.6.0.0/22        24      9583       0       192.168.122.1/3323
```



- Notice that the subnet mask and “MaxLen” in this table match on most entries. So for one of these prefixes (such as 1.0.0.0/24) this means that if anyone advertised this prefix with GREATER than a /24 mask (like /27 or /30) it would automatically be classified as “Invalid”.
- Look at the last prefix of 1.6.0.0/22. The MaxLen on this one is /24 which means that any prefix of 1.6.0.x from a /22 up to a /24 will be valid IF the originating ASN is 9583. Any prefix of 1.6.0.x that is GREATER than /24 will be Invalid.

Verifying Connection to RPKI Validator

```
CSR1#show bgp rpki server
% Command accepted but obsolete, unreleased or unsupported; see documentation.

BGP SOVC neighbor is 192.168.122.1/3323 connected to port 3323
Flags 64, Refresh time is 120, Serial number is 41, Session ID is 42198
InQ has 0 messages, OutQ has 0 messages, formatted msg 124
Session ID flags 3, Session flags 4008
Neighbor Statistics:
  Prefixes 472969
  Connection attempts: 1
  Connection failures: 0
  Errors sent: 0
  Errors received: 0

Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Minimum incoming TTL 0, Outgoing TTL 255
Local host: 192.168.122.68, Local port: 28978
Foreign host: 192.168.122.1, Foreign port: 3323
Connection tableid (VRF): 0
Maximum output segment queue size: 50
```



- SOVC = Source of Value object Collection

BGP ROV Caveats

- + BGP ROV only works when an incoming prefix matches an existing prefix in the RPKI Repository
- + Currently only about ½ of all BGP global prefixes have been registered as ROAs
- + A malicious prefix that is not in the RPKI Repo can still cause a BGP Hijack attack

<https://rpki.cloudflare.com/?view=statistics>



Assisting iBGP Peers

- + All routers within an ASN do not need to be configured for ROV
- + One router can pass it's ROA state to its iBGP peers
- + This method utilizes BGP extended communities

```
router bgp 1
  bgp log-neighbor-changes
  bgp rpki server tcp 192.168.122.1 port 3323 refresh 120
  neighbor 7.1.7.7 remote-as 7
  neighbor 9.1.9.9 remote-as 1
  neighbor 9.1.9.9 send-community extended
  neighbor 9.1.9.9 announce rpki state
```



- ROV = Route Origin Validation

ROV Extended Communities

- + RFC 8097 defines structure of BGP Extended Community for "Prefix Origination Validation State"
- + All begin with 0x4300:0:X where "X" identifies validation state.

Value	Meaning
0	Lookup result = "valid"
1	Lookup result = "not found"
2	Lookup result = "invalid"

```
R9#show ip bgp 1.7.184.0/24
```

```
7.1.7.7 from 9.1.9.1 (7.1.7.1)  
Origin incomplete, metric 0, localpref 100, valid, internal  
Extended Community: 0x4300:0:0  
rx pathid: 0, tx pathid: 0
```

```
R9#show ip bgp 155.155.155.0/24
```

```
7.1.7.7 from 9.1.9.1 (7.1.7.1)  
Origin incomplete, metric 0, localpref 100, valid, internal  
Extended Community: 0x4300:0:1  
rx pathid: 0, tx pathid: 0
```



Modifying RPKI Validation Behavior

```
CSR1(config-router)#bgp bestpath prefix-validate ?  
  allow-invalid  Allow invalid routes to be considered for bestpath  
  disable       Disable prefix validation
```

- + Allow-Invalid
 - + Overrides default behavior of preventing Invalid prefix installment into the IP Routing Table
 - + Allows you to set Route-Map behavior on Invalid prefixes
- + Disable
 - + Downloads RPKI ROA information but doesn't use it in any way
 - + Can be used if you want to validate connection and ROA download between router and RPKI Validator but not actually act on received information



