DE-CIX Academy: BGP Security

Handout Document

Where
networks
meet

Notice of Liability

Despite careful checking of content, we accept no liability for the content of external links. Content on linked sites is exclusively the responsibility of the respective website operator.

Links visited during the webinar

- **RFCs:**
  - **RFC7454 - BGP Operations and Security (if you only read one, read this one)**
  - RFC2385 - TCP MD5 Protection
  - RFC5925 - TCP Authentication Option
  - RFC5082 - Generalized TTL Security Mechanism
  - RFC1918 - IPv4 prefixes for private networks
  - RFC5737 - IPv4 prefixes for documentation
  - RFC3849 - IPv6 prefix for documentation
  - RFC5398 - AS numbers reserved for documentation
  - RFC6996 - AS numbers reserved for private use
  - RFC7300 - Last AS reservation
  - RFC6480 - RPKI Introduction
- **Other links:**
  - IANA IPv4 Special Registry
  - IANA IPv6 Address Space
  - IANA Special-Purpose Autonomous System Numbers
  - RIPE NCC - RPKI Documentation
  - MANRS - Mutually Agreed Norms for Routing Security
- **Tools:**
  - Regular Expression Tester
  - Numerical Regular Expression Generator
  - RPKI Validator
- **More DE-CIX Acacemy**
  - Visit our website for recorded webinars and more:
    - https://www.de-cix.net/en/about-de-cix/academy/videos-and-webinars
    - https://www.de-cix.net/en/about-de-cix/academy/white-papers

DE-CIX Academy: BGP Security

Handout Document

Where networks meet

Security Measures presented in the webinar

Measures in **bold** are **highly recommended**

- → **Simple measures**
  - **Set a reasonable high maximum prefix on each eBGP session**
  - Implement MD5 password protection on most important sessions
  - **Enable TTL Security**
- → **Prefix filtering (IPv4 and IPv6)**
  - **filter against private networks**
  - **filter against reserved networks**
  - **filter against IXP peering LANs**
  - **filter against your own prefixes**
  - **filter against your customers prefixes (caution if customers are multi-homed)**
- → **AS path filtering**
  - **filter against private AS numbers in the path (anywhere)**
  - **filter against reserved AS numbers in the path (anywhere)**
  - **filter against other reserved AS numbers**
    - A list to be filtered can be found here (except AS112, you should allow that one)

Configuration Example (Cisco)

Prefix-list for unwanted prefixes:

```
 ip prefix-list ipv4-unwanted permit 192.168.0.0/16 le 32
 ip prefix-list ipv4-unwanted permit 172.16.0.0/12 le 32
 ip prefix-list ipv4-unwanted permit 10.0.0.0/8 le 32
 ip prefix-list ipv4-unwanted permit 224.0.0.0/4 le 32
 ip prefix-list ipv4-unwanted permit 240.0.0.0/4 le 32

 ipv6 prefix-list ipv6-unwanted deny 2000::/3 le 48
 ipv6 prefix-list ipv6-unwanted permit ::0/0 le 128
```

Prefix-list for too large and too small prefixes:

```
 ip prefix-list ipv4-unwanted permit 0.0.0.0/0 ge 25
 ip prefix-list ipv4-unwanted permit 0.0.0.0/0 ge 1 le 7

 ipv6 prefix-list ipv6-unwanted permit ::0/0 ge 49
 ipv6 prefix-list ipv6-unwanted permit ::0/0 ge 1 le 19
```

Prefix-list for IXP LANs (Example: Peering LAN DE-CIX Frankfurt (a /21)):

```
 ip prefix-list ipv4-unwanted permit 80.81.192.0/21 le 32
 ipv6 prefix-list ipv6-unwanted permit 2001:7f8::/64 le 128
```

Route-map, using above prefix-list

```
 route-map peering-in deny 50
   match ip address prefix-list ipv4-unwanted
   match ipv6 address prefix-list ipv6-unwanted
```

If you use the same route-map for IPv4 and IPv6 you **must** have **two match statements** (one for v6 and one for v6) in the **same** entry.

AS-Path filter list for unwanted ASes in the path:

We do not want:

→ AS64496 - AS64511 (Documentation ASes)

→ AS64512 - AS65534 (Private ASes)

→ AS65535 (Reserved)

→ AS65536 - AS65551 (Documentation ASes)

- You can summarize this with AS64496 - AS65551
- And split up again in
  - 64496 - 64499
  - 64500 - 64999
  - 65000 - 65499
  - 65500 - 65549
  - 65550 - 65551
- And convert these into regular expressions:
  - _6449[6-9]_
  - _64[5-9][0-9][0-9]_
  - _65[0-4][0-9][0-9]_
  - _655[0-4][0-9]_
  - _6555[01]_

→ So the config looks like:

```
ip as-path access-list 99 permit _6449[6-9]_
ip as-path access-list 99 permit _64[5-9][0-9][0-9]_
ip as-path access-list 99 permit _65[0-4][0-9][0-9]_
ip as-path access-list 99 permit _655[0-4][0-9]_
ip as-path access-list 99 permit _6555[01]_

route-map peering-in deny 60
  match as-path 99
```