BGP configuration best practices

Document produced by ANSSI (French National Security Agency of Information Systems), in collaboration with the following operators:

- Association Kazar ;
- France-IX ;
- Jaguar Network ;
- Zayo France (formerly Neo Telecoms) ;
- Orange;
- RENATER ;
- SFR.

Document formatted using LaTeX. Figures produced using the TikZ tool.

You may send any comments and remarks to the following address:

guide.bgp@ssi.gouv.fr

Table of contents

Introduction

This document, created with the co-operation of French operators, is intended to present and describe good configuration practices for the BGP[1] routing protocol. It is intended first and foremost for BGP router administrators, as well as for those familiar with the BGP deployment architectures. Readers who would like information about the BGP protocol may refer to the report from the French observatory of Internet resilience.

The configuration elements presented in this document apply to the EBGP[2] sessions, i.e. sessions established between two distinct ASes. Each best practice is accompanied by different implementation configuration examples. The following table indicates the routers and operating system versions used.

Operating system	Version used
SR-OS (Alcatel-Lucent)	10.0r5
IOS (Cisco)	15.2(4)S
Junos (Juniper)	11.4R3.7
OpenBGPD (OpenBSD)	5.3

Routers and operating systems used for the configuration examples.

The configuration examples provided have all been tested on the indicated implementations. These extracts are provided for information purposes only: they should be adapted to the deployment environment. ANSSI[3] (French Network and Information Security Agency) declines all responsibility as to the consequences from the use of these examples.

[1] Border Gateway Protocol.
[2] External Border Gateway Protocol.
[3] Agence nationale de la sécurité des systèmes d'information.

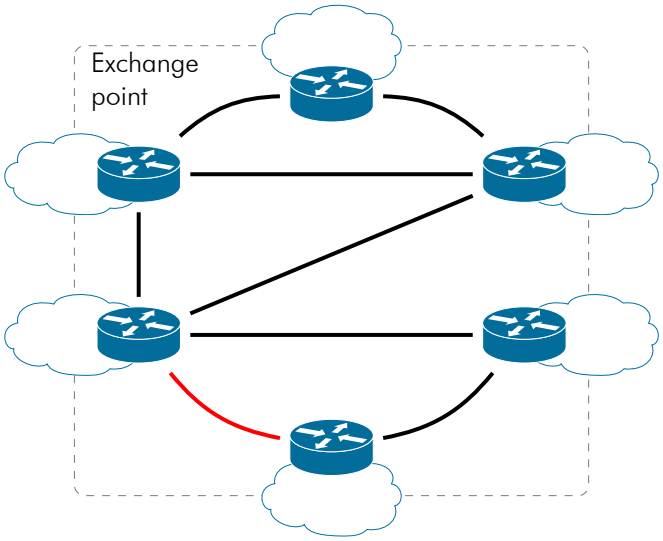
Chapter 1

Configuration recommendations

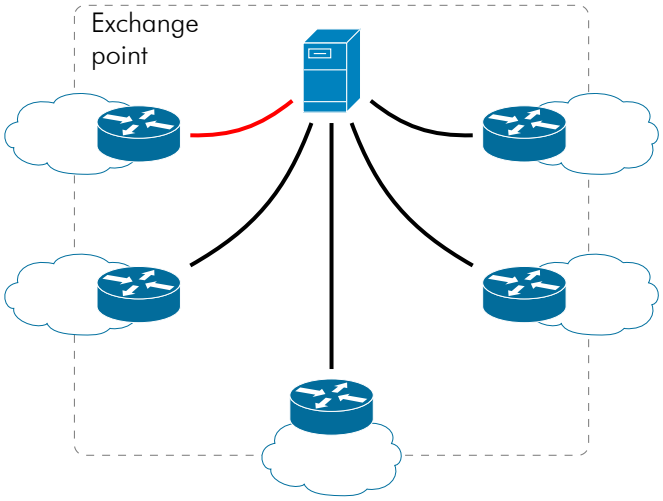
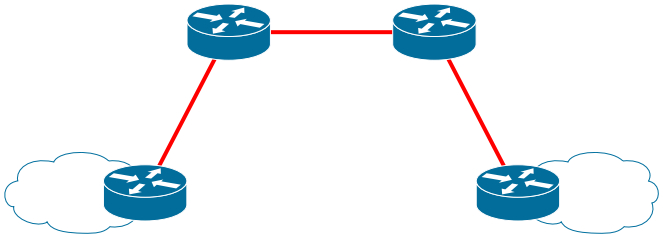
This chapter brings together all the configuration best practices mentioned in this document and gives the associated recommendation levels. The types of interconnection and relationship between ASes concerned by these best practices are explained in the following sections.

1.1 Types of interconnection

The following table describes the types of interconnection targeted by the configuration recommendations. The red link in each figure represents the interconnection described.

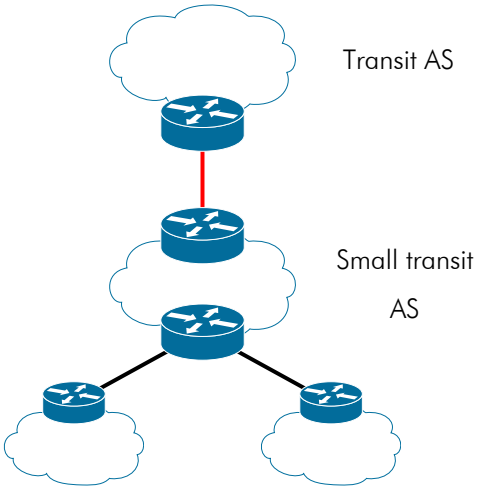
Description	Diagram
Interconnection 1: bilateral peering in an Internet exchange point. This type of interconnection is established using an equipment (such as a switch) managed by the exchange point (not shown on the diagram). Each AS establishes one or more sessions with one or more other ASes.	

[1] *peering*: agreement between peers where each one declares the prefixes it manages.

Interconnection 2: peering using a route server in an exchange point. This type of interconnection enables peers connected to a route server to receive every route declared by the other peers.	
Interconnection 3: private peering between two ASes in a Network Access Point, or interconnection in a telecommunications room. This type of interconnection is performed using a point-to-point link between two peers.	
Interconnection 4: session established in multihop. The interconnection between the BGP routers is not direct, and is established over a routed network such as the Internet.	

1.2 Types of relationship between ASes

The following table describes the types of relationship between ASes mentioned throughout this document. The red link in each figure represents the relationship described.

Description	Diagram
Relationship 1: transit / stub customer. This type of relationship exists between a transit AS and a stub AS which does not offer a transit service.	
Relationship 2: transit AS / small transit AS. This type of relationship exists between a transit AS and a customer AS. This customer AS is also a transit AS for one or more ASes.	

Relationship 3: peering. This type of relationship exists between two ASes which exchange prefixes, without one of these ASes providing the other with a transit service.	

1.3 Recommendations

The recommendation levels which apply to a given configuration element are defined on a three-star scale:

: desirable

: recommended

: highly recommended

1.3.1 Recommendations depending on the type of interconnection

The application of the following configuration elements depends on the interconnection types. Links to the different sections of the document are indicated between parentheses for each best practice.

Best practices	Interconnections	Recommendation levels	Remarks
TCP MD5 (2.1)	Interconnections 1 and 4	★★★	The use of this mechanism is strongly recommended on non-dedicated interconnections.
	Interconnection 2	★★☆	
	Interconnection 3	★☆☆	
Filtering on the peer's AS number (3.7)	Interconnections 1, 3 and 4	★★★	Systematic filtering on the peer AS number.

1.3.2 Recommendations depending on the type of relationship between ASes

The application of the following configuration elements depends on the relationships between the ASes. A dash indicates that the recommendation does not apply to the peer.

<table>
<tr><th>Best practices</th><th>Types of relationship</th><th>Recommendation levels</th><th>Remarks</th></tr>
<tr><td rowspan="5">Filtering of the prefixes assigned to a peer (3.2)</td><td rowspan="2">Relationship 1</td><td>Transit AS side: ★★★</td><td rowspan="5">Systematic filtering for a stub AS.</td></tr>
<tr><td>Customer side: –</td></tr>
<tr><td rowspan="2">Relationship 2</td><td>Transit AS side: ★★☆</td></tr>
<tr><td>Customer side: –</td></tr>
<tr><td>Relationship 3</td><td>★★☆</td></tr>
<tr><td rowspan="3">Limiting the maximum number of prefixes accepted from a peer (3.6)</td><td rowspan="2">Relationships 1 and 2</td><td>Transit AS side: ★★★</td><td rowspan="2">Filtering to be implemented by the transit AS.</td></tr>
<tr><td>Customer side: –</td></tr>
<tr><td>Relationship 3</td><td>★★★</td><td>Filtering to be implemented by each peer.</td></tr>
</table>

Private AS number removal (3.5)	All types of relationship	★★★	The private AS numbers should be systematically trimmed.

1.3.3 General recommendations

The following configuration elements are applicable regardless of the interconnection types and the relationships between AS.

Best practices	Recommendation levels	Remarks
Martians filtering (3.1)	★★★	Systematic filtering.
Filtering of the prefixes that are too specific (3.3)	★★★	More specific than /24 for IPv4 (RIPE-399 [1]) and /48 for IPv6 (RIPE-532 [2]).
Default route filtering (3.4)	★★★	Systematic filtering if the default route does not need to be advertised (except by explicit request from a customer).
Logging (4.1)	★★★	Logging of the adjacency changes on each router and event notification for monitoring.

Graceful Restart (4.2)		This mechanism is used to strengthen the interconnection robustness. When the BGP process is restarting, packets are still forwarded using previously learned routes.

Chapter 2

Session security

The current BGP version specifications (version 4) do not define a mechanism to protect the sessions. As the BGP protocol is supported by TCP, the sessions may be terminated by sending TCP RST packets, which may enable an attacker to perform a denial of service [3, 4, 5]. Although the implementation of this type of attack implies certain pre-requisites, TCP MD5 is a mechanism which complements the other security mechanisms and whose use is part of defence-in-depth approach.

2.1 Message authentication

RFC 4271 [6], published in January 2006, specifies that the BGP implementations should enable use of the authentication mechanism provided by the TCP option commonly called TCP MD5, and described in RFC 2385 [7]. This mechanism is available in most BGP implementations and enables the TCP message integrity and authenticity to be guaranteed by including a MAC[1] calculated using the MD5 hash function.

The introduction of this mechanism is based on a secret shared between two routers. The algorithm applies to the following elements:

- A pseudo IP header comprising the source IP address, the destination IP address, the protocol number and the segment length;
- The TCP header, except for the options, with a null value for the checksum;
- The TCP segment data.

The segment recipient calculates the MAC in the same way and checks if the result is the same as the value contained in the TCP MD5 option. In the event of a failure, the segment is silently rejected. If the secret changes during a session, the packets transmitted by the peer that kept the old secret are rejected and the session expires once the hold time is exceeded.

TCP MD5 is not a robust cryptographic mechanism. In particular, this mechanism does not conform to the ANSSI recommendations. However, the existing implementations at the writing of this document do not propose the TCP Authentication Option defined in RFC 5925 [8], which should enable other algorithms to be used. Despite its obsolescence, TCP MD5 constitutes an additional security element regarding best configuration practices. In the absence of a more robust mechanism, TCP MD5 should be

[1] Message Authentication Code.

used systematically when the BGP interconnection is performed in multi-hop, or using a shared support (for example a switch) within an exchange point. When the interconnection is performed between two routers that propose a more robust cryptographic mechanism, it must be used in place of TCP MD5.

A different secret must be configured for each interconnection. The secret used should be strong, or else there is no point in the mechanism provided by TCP MD5. A secret's strength depends on its length and its character classes.

TCP MD5 - Alcatel-Lucent routers

Sample 2.1 : Command allowing the configuration of TCP MD5 authentication

```
neighbor <ip-address> authentication-key <secret>
```

Sample 2.1 - Comments

This sample shows how to configure TCP MD5 authentication for a given peer on an Alcatel-Lucent router using the `authentication-key` command. The key (`secret`) is a character string known by both peers .

Sample 2.2 : TCP MD5 authentication configuration example

```
neighbor 192.0.2.3 authentication-key ght8CD%E7am
```

TCP MD5 - Cisco routers

Sample 2.3 : Command allowing the configuration of TCP MD5 authentication

```
Cisco(config-router)#neighbor <ip-address> password <string>
```

Sample 2.3 - Comments

TCP MD5 authentication can be configured for a given peer (identified by an IP address). The secret is a character string known by both peers.

Sample 2.4 : TCP MD5 authentication configuration example

```
Cisco(config)#router bgp 64506
Cisco(config-router)#neighbor 192.0.2.3 password ght8CD%E7am
```

TCP MD5 - Juniper routers

Sample 2.5 : TCP MD5 authentication configuration example

```
[edit protocols bgp group session-to-AS64506 neighbor
   192.0.2.6]
root@Juniper# set authentication-key ght8CD%E7am
```

Sample 2.5 - Comments

This sample shows how to configure TCP MD5 authentication on a Juniper router using the command `set authentication-key`. The key is a character string shared by both peers.

TCP MD5 - OpenBGPD routers

Sample 2.6 : Command allowing the configuration of TCP MD5 authentication

```
tcp md5sig {password | key} <secret>
```

Sample 2.6 - Comment

The secret can be an ASCII character string (keyword `password`) or a hexadecimal string (keyword `key`).

Sample 2.7 : TCP MD5 authentication configuration example

```
tcp md5sig password "ght8CD%E7am"
```

Chapter 3

Prefix filtering

BGP does not provide a mechanism to validate the prefix advertisements. So, an AS may advertise any prefix. These may be prefixes which are not managed by the AS (prefix hijacking) or prefixes which should not be advertised within the Internet. This section presents different filtering rules and methods to limit the spread of illegitimate advertisements.

3.1 Reserved prefix filtering

Martians are prefixes that are reserved for specific purposes. For example, these may be private address blocks defined in RFC 1918 [9] and in RFC 6890 [10]. The martians should not be advertised throughout the Internet and therefore constitute a first category of prefixes that should be filtered. The filters on these prefixes must be applied to both incoming and outgoing advertisements.

The IANA[1] maintains a list of reserved IPv4 prefixes [11], for which the version of 22nd May 2013 is provided [2] in table 3.1. This table also contains the `224.0.0.0/4` prefix, which is reserved for the multicast. In addition, the IANA maintains a list of reserved IPv6 prefixes [13]. Table 3.2 shows the version of 1st May 2013 with the `ff00::/8` prefix, which is reserved for IP multicasting. The list also contains the following prefixes: `fc00::/7` (unique local), `fe80::/10` (link-local), the more specific prefixes than `2002::/16` (reserved for the 6to4 protocol), and `2001:db8::/32` (reserved for documentation). You may refer to the following documents, which are available online, to establish a list of IPv6 prefixes to be filtered:

- IANA IPv6 Special Purpose Address Registry [13] ;
- Internet Protocol Version 6 Address Space [14] ;
- IPv6 Global Unicast Address Assignments [15].

The IANA allocates prefixes to the RIR[3] which only come from the `2000::/3` prefix, which corresponds to the Global Unicast addresses [16]. At the time of writing, the

[1] Internet Assigned Numbers Authority.

[2] The `192.0.0.0/29` prefix, which is reserved for Dual-Stack Lite [12], as well as prefixes `192.0.0.170/32` and `192.0.0.171/32`, which are reserved for the discovery of NAT64/DNS64, do not appear explicitly in this table: they are included in the `192.0.0.0/24` prefix. In addition, the prefix reserved for the 6to4 relays (`192.88.99.0/24`) is not mentioned in this table.

[3] Regional Internet Registry.

block has not been fully allocated. Tables A.1 and A.2 in appendix A indicate the prefixes reserved on 15th February 2013. The lists of reserved prefixes change over time. Consequently, if the blocks that are not allocated from the 2000::/3 prefix are filtered, the filters based on these lists must be kept up-to-date.

The following examples indicate how to configure filters for the martians. For the sake of brevity, only the Alcatel-Lucent router configuration example is exhaustive.

Reserved IPv4 prefixes	
0.0.0.0/8	reserved for the initialization procedure by which the host learns its own IP address [17] [4]
127.0.0.0/8	reserved for the local loop [17]
169.254.0.0/16	reserved for the local link [18]
198.18.0.0/15	reserved for network equipment performance tests [19]
192.0.0.0/24	reserved for future allocations dedicated to IETF[5] protocols [10]
10.0.0.0/8 172.16.0.0/12 192.168.0.0/16	reserved for private use [9]
192.0.2.0/24 198.51.100.0/24 203.0.113.0/24	prefixes for TEST-NET-1, TEST-NET-2 et TEST-NET-3, reserved for documentation [20]
100.64.0.0/10	reserved for Carrier-Grade NAT [21]
224.0.0.0/4	reserved for IP *multicasting* [22]
240.0.0.0/4	reserved for future use [23]
255.255.255.255/32	limited broadcast: the packets sent to this address are not forwarded by the routers [24]

Table 3.1 Reserved IPv4 prefixes.

[4] According to RFC 1122, this prefix should not be used, except as a source address during an initialisation procedure where the host learns its IP address.

[5] Internet Engineering Task Force.

Reserved IPv6 prefixes	
::1/128	reserved for the local loop [16]
::/128	reserved for the unspecified address [16]
::ffff:0:0/96	reserved for IPv4-Mapped IPv6 addresses [16]
100::/64	reserved for black-holing[6] [25]
2001::/23	reserved by the IANA for protocols (TEREDO for example) [26]
2001::/32	reserved for TEREDO [27]
2001:2::/48	reserved for network equipment performance tests [28]
2001:10::/28	reserved for ORCHID [29]
2001:db8::/32	reserved for documentation [30]
2002::/16 (only more specific prefixes)	reserved for 6to4 [31]
fc00::/7	reserved for Unique-Local addresses [32]
fe80::/10	reserved for Link-Scoped Unicast addresses [16]
ff00::/8	Multicast address range [16]

Table 3.2 Reserved IPv6 prefixes.

Reserved prefixes filtering - Alcatel-Lucent routers

Sample 3.1 : IPv4 reserved prefixes static filter example

```
>config>router>policy-options#
            prefix-list "v4-martians"
                prefix 0.0.0.0/8 longer
                prefix 127.0.0.0/8 longer
                prefix 169.254.0.0/16 longer
                prefix 198.18.0.0/15 longer
                prefix 192.0.0.0/24 longer
                prefix 10.0.0.0/8 longer
                prefix 172.16.0.0/12 longer
```

[6]Black-holing seeks to discard traffic based on its destination or source.

```
                prefix 192.168.0.0/16 longer
                prefix 192.0.2.0/24 longer
                prefix 198.51.100.0/24 longer
                prefix 203.0.113.0/24 longer
                prefix 100.64.0.0/10 longer
                prefix 224.0.0.0/4 longer
                prefix 240.0.0.0/4 longer
                prefix 255.255.255.255/32 exact
            exit
            policy-statement "reject-martians"
                entry 10
                    from
                        prefix-list "v4-martians"
                    exit
                    action reject
                    exit
                exit
                default-action accept
            exit
```

Sample 3.2 : Applying the filter (3.1)

```
>config>router>bgp#
    group "EBGP"
        import "reject-martians"
        export "reject-martians"
        neighbor 192.0.2.3
        exit
    exit
```

Sample 3.3 : IPv6 martians static filter example

```
>config>router>policy-options#
    prefix-list "v6-martians"
        prefix ::1/128 exact
        prefix ::/128 exact
        prefix ::ffff:0.0.0.0/96 longer
        prefix 100::/64 longer
        prefix 2001::/23 longer
        prefix 2001:db8::/32 longer
        prefix 2002::/16 prefix-length-range 17-128
```

```
        prefix fc00::/7 longer
        prefix fe80::/10 longer
        prefix ff00::/8 longer
        prefix 3ffe::/16 longer
        prefix 5f00::/8 longer
    exit
    prefix-list "v6-authorized"
        prefix 2000::/3 prefix-length-range 3-48
    exit
    policy-statement "reject-v6-martians"
        entry 10
            from
                prefix-list "v6-martians"
            exit
            action reject
            exit
        exit
        entry 20
            from
                prefix-list "v6-authorized"
            exit
            action accept
            exit
        exit
        default-action reject
    exit
```

Samples 3.1, 3.2 and 3.3 - Comments

Samples 3.1 and 3.3 give examples of static filter configuration for reserved prefixes (IPv4 and IPv6). These filters are applied to one peer or more, as shown in sample 3.2.

Reserved prefixes filtering - Cisco routers

Sample 3.4 : Creating a prefix-list

```
Cisco(config)#ip prefix-list <list-name> | <list-number> [seq
   number] {deny <network>/<length> | permit <network>/<length
   >} [ge ge-length] [le le-length]
```

Sample 3.4 - Comments

Here are the settings and options available for this command :

- `list-name` and `list-number` identify the *prefix-list* by name or by number;
- `seq number` sets a sequence number between 1 and 2^{32} - 2 which indicates the processing order for the entry. If no sequence number is given, a default number is set. If it is the first entry of the *prefix-list*, the sequence number is 5. For subsequent entries, the number is incremented by 5;
- `deny` and `permit` enable rejecting or allowing a route for a given prefix, respectively;
- the optional parameters `ge ge-length` and `le le-length` can be used to indicate a mask length for which the test is true. The ge keyword means "greater than or equal", and the `le` keyword means "less than or equal".

Sample 3.5 : IPv4 reserved prefixes static filter example

```
Cisco(config)#ip prefix-list ipv4-martians seq 5 deny
   0.0.0.0/8 le 32
Cisco(config)#ip prefix-list ipv4-martians seq 10 deny
   127.0.0.0/8 le 32
Cisco(config)#ip prefix-list ipv4-martians seq 15 deny
   169.254.0.0/16 le 32
Cisco(config)#ip prefix-list ipv4-martians seq 20 deny
   198.18.0.0/15 le 32
Cisco(config)#ip prefix-list ipv4-martians seq 25 deny
   192.0.0.0/24 le 32
Cisco(config)#ip prefix-list ipv4-martians seq 30 deny
   10.0.0.0/8 le 32
Cisco(config)#ip prefix-list ipv4-martians seq 35 deny
   172.16.0.0/12 le 32
Cisco(config)#ip prefix-list ipv4-martians seq 40 deny
   192.168.0.0/16 le 32
Cisco(config)#ip prefix-list ipv4-martians seq 80 deny
   255.255.255.255/32
Cisco(config)#ip prefix-list ipv4-martians seq 500 permit
   0.0.0.0/0 le 24
```

Sample 3.6 : Applying the prefix-list defined in sample 3.5 to a peer on both incoming and outgoing advertisements

```
Cisco(config-router-af)#neighbor 192.0.2.3 prefix-list
   ipv4-martians in
Cisco(config-router-af)#neighbor 192.0.2.3 prefix-list
   ipv4-martians out
```

Sample 3.6 - Comments

The *prefix-list* must be applied to one peer or more. Sample 3.6 shows the configuration of a *prefix-list* filter for both incoming and outgoing advertisements.

Sample 3.7 : IPv6 reserved prefixes static filter example

```
Cisco(config)#ipv6 prefix-list ipv6-filter deny ::1/128
Cisco(config)#ipv6 prefix-list ipv6-filter deny ::/128
Cisco(config)#ipv6 prefix-list ipv6-filter permit 2002::/16
Cisco(config)#ipv6 prefix-list ipv6-filter deny 2002::/16 le
   128
Cisco(config)#ipv6 prefix-list ipv6-filter deny 3FFE::/16 le
   128
Cisco(config)#ipv6 prefix-list ipv6-filter deny 5F00::/8 le
   128
Cisco(config)#ipv6 prefix-list ipv6-filter permit 2000::/3 le
   48
Cisco(config)#ipv6 prefix-list ipv6-filter seq 500 deny ::/0
   le 128
```

Sample 3.7 - Comments

For IPv6 prefixes, the filters can be configured using the `ipv6 prefix-list` command, as shown in sample 3.7. The application of a *prefix-list* to a peer is performed in the same way as for IPv4 (see sample 3.6).

Reserved prefixes filtering - Juniper routers

Sample 3.8 : Filter definition (policy-statement) for IPv4 reserved prefixes

```
[edit policy-options policy-statement ipv4-martians]
root@Juniper# set from route-filter 0.0.0.0/8 orlonger
```

Sample 3.9 : Action definition for the filter ipv4-martians

```
[edit policy-options policy-statement ipv4-martians]
root@Juniper# set then reject
```

Sample 3.10 : IPv4 reserved prefixes filter (non exhaustive)

```
[edit policy-options]
root@Juniper# show policy-statement ipv4-martians
from {
    route-filter 0.0.0.0/8 orlonger;
    route-filter 127.0.0.0/8 orlonger;
    route-filter 169.254.0.0/16 orlonger;
    route-filter 192.168.0.0/16 orlonger;
    route-filter 192.0.2.0/24 orlonger;
    route-filter 240.0.0.0/4 orlonger;
    route-filter 255.255.255.255/32 exact;
}
then reject;
```

Sample 3.11 : Applying the filter ipv4-martians from sample 3.10

```
[edit protocols bgp]
root@Juniper# set group session-to-AS64502-v4 import
   ipv4-martians
root@Juniper# show group session-to-AS64502-v4
type external;
import ipv4-martians;
peer-as 64502;
neighbor 192.0.2.2;
```

Sample 3.12 : IPv6 reserved prefixes filter (non exhaustive)

```
[edit policy-options]
root@Juniper# show policy-statement ipv6-martians
from {
    family inet6;
    route-filter ::1/128 exact;
    route-filter ::/128 exact;
    route-filter 2001:0000::/23 orlonger;
    route-filter 2001:db8::/32 orlonger;
    route-filter 2002::/16 exact next policy;
    route-filter 2002::/16 longer;
}
then reject;
```

Samples 3.8, 3.9, 3.10, 3.11 and 3.12 - Comments

Samples 3.8 and 3.9 show how to build the `policy-statement` *ipv4-martians*. Sample 3.8 shows the definition of the rules, and sample 3.9 indicates the action to be taken. Sample 3.10 shows a filter that rejects IPv4 *martians*. Sample 3.11 indicates how to apply the filter to a given BGP neighbor. Similarly, it is possible to filter IPv6 reserved prefixes, as shown in sample 3.12.

Reserved prefixes filtering - OpenBGPD routers

Sample 3.13 : IPv4 and IPv6 static filter configuration example

```
# Martians IPv4
deny from any prefix 0.0.0.0/8 prefixlen >= 8
deny from any prefix 127.0.0.0/8 prefixlen >= 8
deny from any prefix 169.254.0.0/16 prefixlen >= 16
deny from any prefix 198.18.0.0/15 prefixlen >= 15

# Martians IPv6
deny from any prefix ::1/128
deny from any prefix ::/128
deny from any prefix ::ffff:0:0/96 prefixlen >= 96
deny from any prefix 64:ff9b::/96 prefixlen >= 96
```

Sample 3.13 - Comments

Sample 3.13 gives an example of static filter configuration for IPv4 et IPv6 reserved prefixes.

3.2 Filtering of the prefixes assigned to a peer

In the case of a BGP session between a transit AS and a stub AS, the customer's prefixes should be filtered by the transit AS in order to drop any illegitimate prefix advertisement.

This type of filtering may be extended to other interconnection types. In the absence of any agreement between the ASes upon the prefixes they advertise, the IRRs[7] should be consulted for the definition of the filters. However, the information they provide may not always be up-to-date. Strict filtering, i.e. filtering which drops any advertisement that does not conform to the declarations in the registries, is therefore not always possible.

The filters on the tested implementations are configured in the same way as the filters presented in section 3.1.

3.3 Filtering too specific prefixes

At the time of writing, the mask length for prefix advertisements should not exceed 24 bits for IPv4 [1] and 48 bits for IPv6[8] [2]. This filtering rule enables the size of the global routing table to be limited.

Filtering too specific prefixes - Alcatel-Lucent routers

Sample 3.14 : Filtering IPv4 prefixes more specific than /24

```
>config>router>policy-options#
    prefix-list "v4-too-specific"
        prefix 0.0.0.0/0 prefix-length-range 25-32
    exit
```

[7] Internet Routing Registries.
[8] Concerning IPv6, this rule may evolve in the future.

Sample 3.15 : Filtering IPv6 prefixes more specific than /48

```
>config>router>policy-options#
    prefix-list "v6-too-specific"
        prefix ::/0 prefix-length-range 49-128
    exit
```

Samples 3.14 and 3.15 - Comments

Samples 3.14 and 3.15 indicate how to filter too specific IPv4 and IPv6 prefixes on an Alcatel-Lucent router. The implementation of these *prefix-lists* is similar to the one from samples 3.1 and 3.2.

Filtering too specific prefixes - Cisco routers

Sample 3.16 : Filtering IPv4 prefixes more specific than /24

```
Cisco(config)#ip prefix-list too-specific seq 5 permit
   0.0.0.0/0 le 24
```

Sample 3.16 - Comments

Sample 3.16 shows how to configure a `prefix-list` in order to discard prefix advertisements more specific than /24. The `prefix-list` is applied to both incoming and outgoing prefix advertisements, as shown in sample 3.6.

Sample 3.17 : Filtering IPv6 prefixes more specific than /48

```
Cisco(config)#ipv6 prefix-list v6-too-specific seq 5 permit
   ::/0 le 48
```

Sample 3.17 - Comments

In a similar way, sample 3.17 shows how to filter IPv6 prefixes more specific than /48. The `prefix-list` is applied to both incoming and outgoing prefix advertisements, as shown in sample 3.6.

Filtering too specific prefixes - Juniper routers

Sample 3.18 : Filtering IPv4 prefixes more specific than /24

```
[edit policy-options policy-statement v4-prefix-filter]
root@Juniper# set term accept-up-to-24 from route-filter
   0.0.0.0/0 upto /24
root@Juniper# set term accept-up-to-24 then next policy
root@Juniper# set then reject
root@Juniper# show
term accept-up-to-24 {
    from {
        route-filter 0.0.0.0/0 upto /24;
    }
    then next policy;
}
then reject;
```

Sample 3.19 : Filtering IPv6 prefixes more specific than /48

```
[edit policy-options policy-statement v6-prefix-filter]
root@Juniper# set term accept-up-to-48 from route-filter ::/0
   upto /48
root@Juniper# set term accept-up-to-48 then next policy
root@Juniper# set then reject
root@Juniper# show
term accept-up-to-48 {
    from {
        route-filter ::/0 upto /48;
    }
    then next policy;
}
then reject;
```

Samples 3.18 and 3.19 - Comments

Sample 3.18 shows how to filter IPv4 prefixes more specific than /48 on a Juniper router. The filtering of IPv6 prefixes is carried out in a similar manner, as shown in sample 3.19.

Filtering too specific prefixes - OpenBGPD routers

Sample 3.20 : Filtering IPv4 prefixes more specific than /24

```
deny from any inet prefixlen > 24
```

Sample 3.21 : Filtering IPv6 prefixes more specific than /48

```
deny from any inet6 prefixlen > 48
```

3.4 Default route filtering

The default route (0.0.0.0/0 for IPv4, and ::/0 for IPv6) should not be advertised, except for a customer who requests it. This avoids accidentally becoming a transit AS, which may lead to very high bandwidth use, and router overload. In addition, the default route should only be accepted by a customer who accesses the Internet via a default route.

Default route filtering - Alcatel-Lucent routers

Sample 3.22 : Default route filtering (IPv4 and IPv6)

```
>config>router>policy-options#
    prefix-list "default-v4"
        prefix 0.0.0.0/0 exact
    exit
    prefix-list " default-v6"
        prefix ::/0 exact
    exit
```

```
    policy-statement "reject-default-v4"
        entry 10
        from
            prefix-list "default-v4"
        exit
        action reject
    exit
    policy-statement "reject-default-v6"
        entry 10
        from
            prefix-list "default-v6"
        exit
        action reject
    exit
```

Sample 3.22 - Comments

Sample 3.22 shows how to filter the IPv4 and IPv6 default routes. The filters can be applied to one peer or more (see sample 3.2).

Default route filtering - Cisco routers

Sample 3.23 : Default route filtering (IPv4 and IPv6)

```
Cisco(config)#ip prefix-list v4-default-route seq 5 deny
   0.0.0.0/0
Cisco(config)#ip prefix-list v4-default-route seq 10 permit
   0.0.0.0/0 le 24
Cisco(config)#ipv6 prefix-list v6-default-route seq 5 deny
   ::/0
Cisco(config)#ipv6 prefix-list v6-default-route seq 10 permit
   ::/0 le 48
```

Sample 3.23 - Comments

On Cisco routers, IPv4 and IPv6 default route filtering can be performed using *prefix-lists*.

Applying the filter to a peer can be done in a similar way as in sample 3.6.

Default route filtering - Juniper routers

Sample 3.24 : Default route filtering (IPv4 and IPv6)

```
[edit policy-options policy-statement no-v4-default-route]
root@Juniper# set term default-route from route-filter
   0.0.0.0/0 exact
root@Juniper# set term default-route then reject

[edit policy-options policy-statement no-v6-default-route]
root@Juniper# set term default-route from route-filter ::/0
   exact
root@Juniper# set term default-route then reject

[edit policy-options]
root@Juniper# show policy-statement no-v4-default-route
term default-route {
    from {
        route-filter 0.0.0.0/0 exact;
    }
    then reject;
}
root@Juniper# show policy-statement no-v6-default-route
term default-route {
    from {
        route-filter ::/0 exact;
    }
    then reject;
}
```

Sample 3.24 - Comment

On Juniper routers, default routes can be filtered using *policy-statements*.

Default route filtering - OpenBGPD routers

Sample 3.25 : Default route filtering (IPv4 and IPv6)

```
deny from any inet prefix 0.0.0.0/0 prefixlen = 0
deny from any inet6 prefix ::/0 prefixlen = 0
```

3.5 Private AS number removal

An AS number that is not unique may be assigned to an organization. For example, a customer AS may be connected to a unique transit AS (by one or more links) which enables it to access the whole Internet. The transit AS then advertises the customer's prefixes. In this case, the transit AS assigns a private AS number to his customer. The private AS numbers extend from 64512 to 65534 [33]. In order to deal with the growing number of ASes, AS numbers over 32 bits have been introduced [34]: the numbers from 4200000000 to 4294967294 are reserved for private use.

The private AS numbers should not be present in advertisements propagated throughout the Internet since they may be used by several ASes at the same time. Outbound filtering to remove the private AS numbers is therefore necessary. Configuration examples are provided for all implementations tested, except for OpenBGPD which does not provide this functionality at the time of writing.

Private AS number removal - Alcatel-Lucent routers

Sample 3.26 : Command enabling private AS number removal

```
>config>router>bgp# remove-private [limited] [skip-peer-as]
```

Sample 3.27 : Private AS number removal in prefix advertisements

```
>config>router>bgp#
    group "EBGP"
        remove-private
        neighbor 192.0.2.3
        exit
    exit
```

Samples 3.26 and 3.27 - Comments

The option *limited* removes all the private AS numbers from the `AS_PATH` to the first non private AS number. The option *skip-peer-as* allows to keep a private AS number if it is the peer's AS number. Sample 3.27 shows how to configure private AS number removal for a peer.

Private AS number removal - Cisco routers

Sample 3.28 : Command enabling private AS number removal

```
Cisco(config-router)#neighbor <ip-address> | <group-name>
   remove-private-as [all [replace-as]]
```

Sample 3.28 - Comments

Here are the settings and options available for this command:

- `ip-address` and `group-name` indicate the peer's address, or the peer group to which the command applies ;
- the keyword `all` enables the removal of all private AS numbers contained in the `AS_PATH`;
- `replace-as` replaces every private AS number with the local AS number (the local AS being the AS to which the router belongs).

Sample 3.29 : Command usage example `remove-private-as`

```
Cisco(config-router)#address-family ipv4
Cisco(config-router-af)#neighbor 192.0.2.3 remove-private-as
```

Sample 3.29 - Comments

Sample 3.29 shows how to use the command `remove-private-as`. In this case, AS64506 advertisements to its peer will not contain private AS numbers. On older IOS versions, the behavior can be different. In particular, in versions prior

to version 15.1(2)T [35], if the `AS_PATH` contains public AS numbers, no private AS number will be removed.

Private AS number removal - Juniper routers

Sample 3.30 : Example of removing private AS numbers in advertisements

```
[edit protocols bgp]
root@Juniper# set group session-to-AS64503 neighbor 2001:db8
   :0:3:fac0:100:22d3:ce80 remove-private
root@Juniper# show group session-to-AS64503
type external;
log-updown;
family inet6 {
    unicast;
}
peer-as 64503;
neighbor 2001:db8:0:3:fac0:100:22d3:ce80 {
    remove-private;
}
```

Sample 3.30 - Comments

The `remove-private` command enables the removal of private AS numbers on Juniper routers. Sample 3.30 shows how to configure private AS numbers removal for a given peer.

3.6 Limiting the maximum number of prefixes accepted from a peer

Filtering on the number of prefixes advertised by a peer is intended to protect the routers from overload. However, this type of filter also helps protecting the routing consistency. For example, a customer AS may advertise by mistake the whole Internet routing table to its transit AS. If this transit AS does not carry out any filtering and accepts these

advertisements, it is highly probable that it will choose the routes advertised by his customer and propagate them to its peers. In fact, for economic reasons, the values of the `LOCAL_PREF` attribute associated with the customer's routes are generally higher than those of other peers' routes. Consequently, following the advertisement of the customer's peers, a certain number of peers may in turn choose these routes as the best ones, making the prefixes inaccessible. This type of incident occurred on several occasions [36].

To prevent re-advertisement of the routing table, it is strongly recommended to apply a filter to the maximum number of prefixes advertised by a customer or an AS with which a peering relationship is established. Equipments usually offer a certain degree of flexibility by enabling the configuration of the number of prefixes advertised from which the session will be shut down, along with the configuration of a threshold from which warning messages can be generated or SNMP traps sent. For example, for a peer that advertises 200 prefixes, it is possible to set a maximum limit of 1000 prefixes and an alert threshold of 400 prefixes.

Filter on the number of prefixes - Alcatel-Lucent routers

Sample 3.31 : Maximum number of prefixes filter configuration

```
>config>router>bgp>group#
# neighbor <address> prefix-limit <limit> [log-only] [
    threshold <percentage>]
```

Sample 3.32 : Maximum number of prefixes filter configuration example

```
# neighbor 192.0.2.3 prefix-limit 1000 threshold 50
```

Sample 3.31 et 3.32 - Comments

Here are the settings and options available for the command shown in sample 3.31:

- `prefix-limit` is the maximum number of prefixes allowed for a given peer;
- `threshold` is the percentage of the maximum number of prefixes from which the router will generate warning messages. When this threshold is reached, a SNMP trap is sent. Once the limit is exceeded, the BGP session

is shut unless the `log-only` option is configured, in which case only a new warning is issued.

Sample 3.32 gives a configuration example of a maximum number of 1000 prefixes, with an alert threshold of 500 prefixes.

Filter on the number of prefixes - Cisco routers

Sample 3.33 : Command enabling maximum number of prefixes filtering

```
Cisco(config-router-af)#neighbor <ip-address> | <group-name>
   maximum-prefix <maximum> [threshold] [restart
   restart-interval] [warning-only]
```

Sample 3.33 - Comments

Here are the settings and options available for this command:

- `maximum` is the maximum number of prefixes allowed for a given peer;
- `threshold` is the percentage of the maximum number of prefixes from which the router will generate warning messages. By default, messages are generated when the threshold of 75 % of the maximum number is exceeded;
- `restart-interval` specifies the time interval, in minutes, after which the session is reestablished (from 1 to 65 535 minutes) ;
- `warning-only` indicates that the session should not be terminated when the number of advertised prefixes exceeds the limit, but that warning messages should be generated instead.

Sample 3.34 : Maximum number of prefixes filter configuration example

```
Cisco(config-router)#address-family ipv6
Cisco(config-router-af)#neighbor 2001:db8:0:3:fac0:100:22d3:
   d000 maximum-prefix 1000 50
```

Sample 3.34 - Comment

In this example, the maximum number of prefixes allowed is 1000, and the router will generate warning messages when 500 or more prefixes are advertised.

Filter on the number of prefixes - Juniper routers

Sample 3.35 : Command enabling maximum number of prefixes filtering

```
prefix-limit {
  maximum <number>;
  teardown <percentage> [idle-timeout {forever} | <minutes>];
}
```

Sample 3.35 - Comments

Here are the settings and options available for this command:

- `maximum` is the maximum number of prefixes allowed for a peer (from 1 to 2^{32} - 1) ;
- `teardown` indicates that the session should be terminated if the maximum number of prefixes is reached. If `teardown` is followed by a percentage, warning messages are logged when this percentage is exceeded. Once the session is shut, it is reestablished after a "short time" [37]. If a duration is specified using the `idle-timeout` keyword, then the session will be reestablished once this duration has elapsed. If `forever` is specified, then the session will not be reestablished.

For the sake of brevity, the different configuration hierachical levels are not shown in these samples.

Sample 3.36 : Maximum number of prefixes filter configuration example

```
[edit protocols bgp]
root@Juniper# set group session-to-AS64503 neighbor 2001:db8
   :0:3:fac0:100:22d3:ce80 family inet6 unicast prefix-limit
   maximum 1000 teardown 50
```

Filter on the number of prefixes - OpenBGPD routers

Sample 3.37 : Command enabling maximum number of prefixes filtering

```
max-prefix <number> [restart <minutes>]
```

Sample 3.37 - Comments

Here are the settings and options available for this command:

- `number` is the maximum number of prefixes allowed. Beyond this threshold, the session is shut ;
- if `restart` is specified, the session will be reestablished after the specified duration (in minutes).

3.7 Filtering on the peer's AS number

In general, advertisements for which the first AS number in the `AS_PATH` (i.e. the AS number the furthest to the left) is not that of the peer should not be accepted. For example, in the case of an interconnection between a transit AS and a stub AS, the customer's advertisements should be filtered in order to eliminate those whose `AS_PATH` contains other AS numbers than the one of the customer.

`AS_PATH` filtering - Alcatel-Lucent routers

Sample 3.38 : Command allowing the creation of an AS_PATH filter rule

```
>config>router>policy-options#
    as-path <"name"> <"regular expression">
```

Sample 3.39 : AS_PATH filtering example

```
>config>router>policy-options#
    as-path "from-AS64506" "64506 .*"
    policy-statement "from-AS64506"
        entry 10
            from
                protocol bgp
```

```
                as-path "from-AS64506"
            exit
            action accept
            exit
        exit
        default-action reject
    exit
```

Sample 3.38 et 3.39 - Comments

Sample 3.38 gives an example of an `AS_PATH` filter. The filter rules are defined using regular expressions. The filter shown in sample 3.39 can be applied to a peer in the same way as indicated in sample 3.2.

`AS_PATH` filtering - Cisco routers

Sample 3.40 : Command allowing to filter the first AS in the AS_PATH

```
Cisco(config-router)#bgp enforce-first-as
```

Sample 3.40 - Comments

The command `bgp enforce-first-as` enables to discard routes whose first AS number in the `AS_PATH` differs from that of the peer advertising these routes. This filtering is enabled by default [35].

Sample 3.40 shows how to make this feature explicitly appear in the router configuration.

AS_PATH filtering - Juniper routers

Sample 3.41 : Command allowing to filter the first AS in the AS_PATH

```
[edit policy-options]
root@Juniper# set as-path from-AS64506 "^64506 .*"

[edit policy-options policy-statement match-peer-AS64506]
root@Juniper# set term peer-AS64506 from as-path from-AS64506
root@Juniper# set term peer-AS64506 then accept
root@Juniper# set term reject-other-peers then reject
root@Juniper# show
term peer-AS64506 {
    from as-path from-AS64506;
    then accept;
}
term reject-other-peers {
    then reject;
}
```

Sample 3.41 - Comments

The rules, based on regular expressions, are created using the command as-path <*name*> <*regular-expression*>.

In this sample, a rule on the AS_PATH is created using the regular expression ^64506 .*. The AS_PATH matching this rule are those whose first AS number is 64506. In the Junos syntax, « . » matches any AS number.

AS_PATH filtering - OpenBGPD routers

Sample 3.42 : AS_PATH filtering example

```
enforce neighbor-as {yes}
```

Sample 3.42 - Comments

Like the example given for Cisco routers (3.40), the command given in sample 3.42 enables to reject routes advertised by an AS whose number is not the last added to the `AS_PATH`. This is the default behavior of the implementation.

Chapter 4

Other BGP configuration elements

4.1 Use of logging

The routers offer numerous logging functions. Logging is used to detect stability problems and therefore, it may become useful during post-incident interventions. The records are used to identify the equipment which was the origin of the log entry, the session concerned, the cause and the exact time and date of the incident. For BGP, and on the Cisco and Juniper routers, the adjacency change events are not logged by default. These events correspond to the session status changes and so must be logged. By default, OpenBGPD logs the status changes using Syslog [38]. For the Alcatel-Lucent routers, the BGP events are logged by default.

The routers also offer more advanced logging functions, for example to save the content of the messages exchanged. These functions may be useful for debugging purposes.

BGP events logging - Alcatel-Lucent routers

Sample 4.1 : Log entries generated by an Alcatel-Lucent router

```
52915 2012/12/25 17:05:17.00 CET MINOR: BGP #2001 vprn300 Peer 12:
    198.51.100.50 "VR 12: Group CE-IPVPN300: Peer 198.51.100.50:
    moved into established state"

52914 2012/12/25 17:04:45.70 CET WARNING: BGP #2002 vprn300 Peer 12:
     198.51.100.50 "VR 12: Group CE-IPVPN300: Peer 198.51.100.50:
    moved from higher state OPENSENT to lower state IDLE due to
    event TCP SOCKET ERROR"

52913 2012/12/25 17:04:45.70 CET WARNING: BGP #2011 vprn300 Peer 12:
     198.51.100.50 "VR 12: CE-IPVPN300: Peer 198.51.100.50: remote
    end closed connection"

52912 2012/12/25 17:04:45.66 CET WARNING: BGP #2005 vprn300 Peer 12:
     198.51.100.50 "VR 12: CE-IPVPN300: Peer 198.51.100.50: sending
    notification: code HOLDTIME subcode UNSPECIFIED"
```

```
52911 2012/12/25 17:04:45.66 CET WARNING: BGP #2002 vprn300 Peer 12:
    198.51.100.50 "VR 12: CE-IPVPN300: Peer 198.51.100.50: moved
   from higher state ESTABLISHED to lower state IDLE due to event
   HOLDTIME"
```

Sample 4.1 - Comments

By default, BGP events are logged in *log 99*: adjacency changes, malformed `UPDATE` messages or `NOTIFICATION` messages. Fine-grained logging configuration is possible, like on Cisco or Juniper routers.

BGP events logging - Cisco routers

Sample 4.2 : BGP adjacency changes logging configuration

```
Router(config-router)#bgp log-neighbor-changes
```

Sample 4.3 : BGP log entries examples

```
Jun 25 11:19:28.111: %BGP-5-ADJCHANGE: neighbor 2001:DB8:0:3:
   FAC0:100:22D3:D000 Up
Jun 25 11:25:37.843: %BGP-4-MAXPFX: No. of prefix received
   from 2001:DB8:0:3:FAC0:100:22D3:D000 (afi 1) reaches 8, max
    10
Jun 25 11:25:37.843: %BGP-3-MAXPFXEXCEED: No. of prefix
   received from 2001:DB8:0:3:FAC0:100:22D3:D000 (afi 1): 11
   exceed limit 10
Jun 25 11:25:37.843: %BGP-5-ADJCHANGE: neighbor 2001:DB8:0:3:
   FAC0:100:22D3:D000 Down BGP Notification sent
```

Sample 4.3 - Comments

This sample gives an example of log entries related to adjacency changes and maximum number of prefixes filtering on a Cisco router. In this example, a BGP session is established with a peer whose IP address is `2001:db8:0:3:fac0:100:22d3:d000`. The second log entry is a warning

message indicating that an alert threshold has been exceeded. The third entry shows that the maximum number of prefixes allowed has been exceeded (11 prefixes advertised, namely one more than the limit set to 10). Finally, the last entry indicates that a `NOTIFICATION` message has been sent to the peer, terminating the session.

BGP events logging - Juniper routers

Sample 4.4 : BGP adjacency changes logging configuration

```
[edit protocols bgp]
root@Juniper# set log-updown
```

Sample 4.5 : BGP log entries examples

```
Jul 15 11:24:07 JUNIPER rpd[1176]: bgp_peer_mgmt_clear:5992:
   NOTIFICATION sent to 192.0.2.1 (External AS 64501): code 6
   (Cease) subcode 4 (Administratively Reset), Reason:
   Management session cleared BGP neighbor
Jul 15 11:24:07 JUNIPER rpd[1176]:
   RPD_BGP_NEIGHBOR_STATE_CHANGED: BGP peer 192.0.2.1 (
   External AS 64501) changed state from Established to Idle (
   event Stop)
Jul 15 11:24:39 JUNIPER rpd[1176]:
   RPD_BGP_NEIGHBOR_STATE_CHANGED: BGP peer 192.0.2.1 (
   External AS 64501) changed state from OpenConfirm to
   Established (event RecvKeepAlive)
```

Samples 4.4 and 4.5 - Comments

On Juniper routers, the command `log-updown` activates adjacency changes logging.

Sample 4.4 gives an example of global activation (for all the BGP peers). Sample 4.5 shows log entries due to a BGP session restart.

BGP events logging - OpenBGPD routers

Sample 4.6 : Log entries generated by OpenBGPD

```
Apr 29 15:58:49 openbsd64-1 bgpd[13682]: neighbor 192.0.2.2: state
    change None -> Idle, reason: None
Apr 29 15:58:49 openbsd64-1 bgpd[13682]: neighbor 192.0.2.2: state
    change Idle -> Connect, reason: Start
Apr 29 15:58:49 openbsd64-1 bgpd[13682]: neighbor 192.0.2.2: state
    change Connect -> OpenSent, reason: Connection opened
Apr 29 15:58:49 openbsd64-1 bgpd[13682]: neighbor 192.0.2.2: state
    change OpenSent -> Active, reason: Connection closed
Apr 29 15:59:54 openbsd64-1 bgpd[13682]: neighbor 192.0.2.2: state
    change Active -> OpenSent, reason: Connection opened
Apr 29 15:59:54 openbsd64-1 bgpd[13682]: neighbor 192.0.2.2: state
    change OpenSent -> OpenConfirm, reason: OPEN message received
Apr 29 15:59:54 openbsd64-1 bgpd[13682]: neighbor 192.0.2.2: state
    change OpenConfirm -> Established, reason: KEEPALIVE message
    received
```

Sample 4.6 - Comments

Sample 4.6 provides a log example generated by OpenBGPD when establishing a session. Adjacency changes events are logged by default in `/var/log/daemon`.

4.2 The Graceful Restart mechanism

The Graceful Restart mechanism, which is specified for BGP in RFC 4724 [39], is used to limit the prefix unavailability due to the BGP process restarting on a router. On a BGP interconnection between two peers, the Graceful Restart capacity declaration is used to keep the packet transfer during the BGP process restart for one of the two routers. The transfer is carried out during a limited time beyond which the routes used are deleted. Once the restart has been performed, the router selects the best routes among the ones sent by its peers and updates its RIB[1] and FIB[2].

[1] Routing Information Base.
[2] Forwarding Information Base.

Graceful Restart - Alcatel-Lucent routers

Sample 4.7 : Graceful Restart configuration on Alcatel-Lucent routers

```
>config>router>bgp>group#
    group "EBGP"
        graceful-restart [stale-routes-time <time>]
```

Sample 4.7 - Comments

The setting `stale-routes-time` determines the maximum time during which the router keeps the routes marked as stale before removing them. This time can take values from 1 to 3600 seconds, the default value being 360 seconds. This mechanism can be configured on a per-neighbor basis, but also for a neighbor group or in the BGP context.

Graceful Restart - Cisco routers

Sample 4.8 : Graceful Restart configuration on Cisco routers

```
Router(config-router)#bgp graceful-restart [restart-time
    <seconds> | stalepath-time <seconds>]
```

Sample 4.8 - Comments

Graceful Restart can be enabled in `router` configuration mode or `address-family` configuration mode. Here are the options available for this command:

- `restart-time` enables to set the maximum time during which the router waits for a peer to restart. This duration can take values from 1 to 3600 seconds, the default value being 120 seconds;
- `stalepath-time` enables to set the maximum time during which the router keeps the routes marked as stale before removing them. This duration can take values from 1 to 3600 seconds, the default value being 360 seconds.

Sample 4.9 : Graceful Restart configuration example

```
Cisco(config)#router bgp 64506
Cisco(config-router)#bgp graceful-restart restart-time 120
Cisco(config-router)#bgp graceful-restart stalepath-time 360
```

Graceful Restart - Juniper routers

Sample 4.10 : Graceful Restart configuration on Juniper routers

```
[edit protocols bgp]
graceful-restart {
  restart-time <seconds>;
  stale-routes-time <seconds>;
}
```

Sample 4.10 - Comments

Here are the options available for this command:

- `restart-time` enables to set the expected duration of a peer restart. The duration can take values from 1 to 600 seconds. The default duration is 120 seconds;
- `stale-routes-time` enables to set the delay during during which the routes marked as stale are kept inside the FIB. The duration can take values from 1 to 600 seconds. The default `stale-routes-time` is 300 seconds.

Sample 4.11 : Graceful Restart configuration example

```
[edit protocols bgp]
root@Juniper# set graceful-restart restart-time 120
root@Juniper# set graceful-restart stale-routes-time 360
root@Juniper# show graceful-restart
restart-time 120;
stale-routes-time 360;
```

Graceful Restart - OpenBGPD routers

OpenBGPD and the Graceful Restart mechanism

The tested version of OpenBGPD does not support the Graceful Restart mechanism. However, OpenBGPD is able to generate the end of RIB marker [39] after the readvertisement of all of its routes to the peer that has restarted. The advertisement of this marker allows the peer to begin the process of route selection, and thus promotes convergence. Without the end of RIB marker, the peer that has restarted must wait some time before starting the selection process.

Chapter 5

General router configuration elements

The mechanisms described in this section are not specific to BGP security but may help strengthen the resistance of the interconnections.

5.1 Preventing IP address spoofing

Denial of service attacks often use spoofed source addresses to hide the origin of the attack and make it harder to set up filters to eliminate this traffic. The URPF (Unicast Reverse Path Forwarding) technique was created to thwart IP address spoofing. This technique is not directly related to BGP but it may be used to limit the impact on a BGP router when hit by a denial of service attack. Its operating principle is based on systematic verification of the correspondence between the source addresses, the input interface on which the packets arrive and the FIB entries that may enable the source to be reached. More precisely, there are three main operating modes, described in RFC 3704 [40]:

- The strict mode is used to check that the source address of a packet which arrives on an interface may be reached by a route present in the FIB. It also checks that the interface which would be used to reach it is the interface on which the packet was received;
- The feasible path mode is an extension of strict mode. In this mode, the alternative routes, i.e. the routes which are not used by the FIB, are also taken into account for the test;
- The loose mode only checks that the source address of a packet which arrives on the router may be reached by a route present in the FIB. The interface which will be used to reach the source is not taken into account for this mode. The loose mode is used to reject the packets whose source IP address is not routed over the Internet.

For these three modes, the packets are dropped when the conditions are not verified.

Strict mode cannot be used with asymmetric routing, as shown in figure 5.1, for it would lead to discard part of the legitimate traffic. For example, on figure 5.1, if URPF is activated in strict mode on the AS A router, the traffic from AS B would be rejected. Indeed, the route taken (from AS B to AS D, then from AS D to AS A) is different from the one used to send traffic to this AS (from AS A to AS C, then from AS C to AS B).

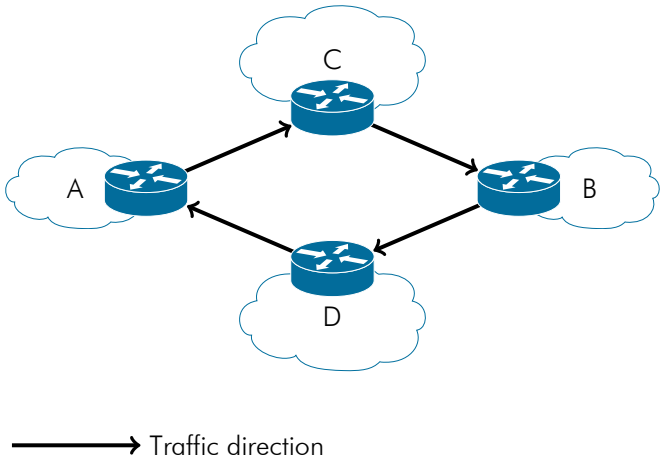


Figure 5.1 Asymmetric routing between AS A and AS B

In this case, it is possible to use the feasible path mode, which takes account of the alternative route via AS D. However, the feasible path mode is implemented on the Juniper routers, but not on the Alcatel-Lucent, Cisco or OpenBGPD routers. For these last implementations, in the case of multihoming, only loose mode may be used.

URPF configuration - Alcatel-Lucent routers

Sample 5.1 : Command enabling URPF

```
config
  router <router-name>
    interface <ip-int-name>
      urpf-check
        mode {strict | loose}
        no mode
      ipv6
        urpf-check
          mode {strict | loose}
          no mode
```

Sample 5.1 - Comments

Sample 5.1 gives the command set to enable URPF on Alcatel-Lucent routers.

By default, this mechanism is not activated.

Here are the settings and options available for this command :

- `mode` activates URPF in strict mode or loose mode ;
- `no mode` activates URPF in strict mode, which is the default mode.

Sample 5.2 : URPF configuration example using loose mode

```
>config>service#
  ies 200 customer 1 create
    interface "from_client" create
      urpf-check
        mode loose
      exit
      ipv6
        urpf-check
          mode loose
        exit
      exit
    exit
```

Sample 5.2 - Comments

Sample 5.2 shows a configuration example of URPF in loose mode on Alcatel-Lucent routers.

If the default route is in the routing table and loose mode is configured, the URPF test will pass. Nevertheless, if the packet source address matches a black-holing route, the URPF test will fail.

By default, unmatched packets are silently discarded.

URPF configuration - Cisco routers

Sample 5.3 : Command enabling URPF on a single interface

```
Cisco(config-if)#ip verify unicast source reachable-via {rx |
   any} [allow-default] [allow-self-ping] [list]
```

Here are the settings and options available for this command:

- `rx` enables *strict* mode, whereas `any` enables *lost* mode;
- `allow-default` includes the default route in the URPF test;
- `allow-self-ping` enables the router to ping its own interfaces, which is blocked by default when URPF is enabled;
- `list` is the *access-list* which will be used if the URPF test fails. It is thus possible to allow exceptions (i.e. source addresses which would make the test fail), or to log incoming packets before discarding them. By default, packets which make the test fail are silently discarded. However, URPF discarded packets counters are updated.

Sample 5.4 : URPF configuration example using loose mode

```
Cisco(config-if)#ip verify unicast source reachable-via any
```

URPF configuration - Juniper routers

Sample 5.5 : Command enabling URPF

```
[edit logical-systems logical-system-name routing-options
   forwarding-table]
[edit routing-instances routing-instance-name instance-type name
   routing-options forwarding-table]
[edit routing-options forwarding-table]
root@Juniper# set unicast-reverse-path (active-paths |
   feasible-paths);
```

Sample 5.6 : Command enabling URPF on a specific interface

```
[edit interfaces interface-name unit logical-unit-number family
    family]
[edit logical-systems logical-system-name interfaces interface-name
    unit logical-unit-number family family]
rpf-check {
  fail-filter <filter-name>;
  mode loose;
}
```

Samples 5.5 and 5.6 - Comments

`unicast-reverse-path` command enables URPF. If the `active-paths` parameter is set, then only actives routes from the FIB will be checked (i.e. chosen routes for packet forwarding). If the `feasible-paths` parameter is set, then alternative routes (i.e. routes that may not be used for packet forwarding, but are present in the RIB) will also be checked by URPF.

Once URPF is configured, the interface where URPF is expected to be used must be configured with the command given in sample 5.6. Like on Cisco routers, a filtering can be set in order to take specific actions (for example, logging if the URPF test fails). By default, packets are silently discarded. Specifying `mode loose` will enable the URPF in *loose* mode.

URPF configuration - PF (OpenBGPD routers)

Sample 5.7 : Command enabling URPF

```
block in [quick] from urpf-failed [label <urpf>]
```

Sample 5.7 - Comments

Sample 5.7 extract shows how to configure URPF using *Packet Filter*. OpenBSD only supports the strict mode.

Moreover, if the default route goes through the interface where URPF is enabled, the route is not excluded from the URPF test, making URPF useless on this interface.

In order to log packets which failed the URPF test, the `log` parameter must be added to the `block` action:
`block in [log] [quick] from urpf-failed [label <urpf>].`

5.2 Hardening the router configuration

The implementation of the configuration best current practices described in this document must be accompanied by router protection measures. More generally, the equipment configurations and the management plane should be hardened. Among other things, you may:

- Use secure protocols to access the router (for example, SSH [41] with public key authentication);
- Restrict access to the equipment:
  - Use of a dedicated administration interface;
  - Connection from authorised IP addresses;
  - Definition of user accounts dedicated to a specific use, etc.;
- Deactivate unused services (processes or protocols);
- Apply configuration best current practices to the different protocols implemented by the equipment;
- Use up-to-date operating systems or firmwares…

The configuration guides offered by equipment manufacturers often provide guidelines related to equipment configuration hardening.

5.2.1 Control plane protection

The tasks carried out in the control plane supply the FIB, i.e. the transfer tables used by the data plane. The routing protocol processes such as BGP should operate within the router control plane. Consequently, control plane protection is also a vital element for BGP security. The varied nature of the tasks carried out in the control plane explains that this plan is controlled by central processing units (CPUs).

However, the data plane is based on ASIC[1] dedicated to specific packet processing (packet transfer operations to an appropriate interface or to the control plane). These hardware components offer very high packet processing capacity, in particular a capacity that is much higher than that of the control plane. Consequently, this control plane is more likely to be overloaded during a denial of service attack than the data plane.

The main objective of control plane protection is the reduction of its attack surface. This involves the implementation of filters in order to discard most of the illegitimate traffic before it reaches the control plane. RFC 6192 [42] describes a number of filters aiming to protect the router control plane and provides configuration examples to implement these filters for Cisco and Juniper routers.

[1] Application Specific Integrated Circuits.

Appendix A

IPv6 addressing space

Tables A.1 and A.2 respectively provide the prefixes reserved by the IETF and the reserved prefixes that belong to 2000::/3. The list may be obtained from the IANA registries: *Internet Protocol Version 6 Address Space* [14] and *IPv6 Global Unicast Address Assignments* [15]. The version of 15th February 2013 was used to generate these tables.

IPv6 reserved address space	
0000::/8 0100::/8 0400::/6 0800::/5 1000::/4 4000::/3 6000::/3 8000::/3 a000::/3 c000::/3 e000::/4 f000::/5 f800::/6 fe00::/9	reserved by the IETF [16]
0200::/7	reserved by IETF [43].
fec0::/10	reserved by IETF [44].

Table A.1 Reserved IPv6 Prefixes.

Global Unicast IPv6 space	
2001:3c00::/22 2d00:0000::/8 2e00:0000::/7 3000:0000::/4	reserved by IANA.
3ffe::/16 5f00::/8	prefixes which were previously reserved for the *6bone*, the IPv6 test network.

Table A.2 *Global Unicast* space.

Bibliography

[1] RIPE-NCC, "RIPE Routing Working Group Recommendations on Route Aggregation." <http://www.ripe.net/ripe/docs/ripe-399>, décembre 2006.

[2] RIPE-NCC, "RIPE Routing Working Group Recommendations on IPv6 Route Aggregation." <http://www.ripe.net/ripe/docs/ripe-532>, novembre 2011.

[3] P. A. Watson, "Slipping in the Window: TCP Reset Attacks, CanSecWest," 2004.

[4] A. Ramaiah, R. Stewart, and M. Dalal, "Improving TCP's Robustness to Blind In-Window Attacks." RFC 5961 (Proposed Standard), Aug. 2010.

[5] J. Touch, "Defending TCP Against Spoofing Attacks." RFC 4953 (Informational), July 2007.

[6] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)." RFC 4271 (Draft Standard), Jan. 2006. Updated by RFCs 6286, 6608, 6793.

[7] A. Heffernan, "Protection of BGP Sessions via the TCP MD5 Signature Option." RFC 2385 (Proposed Standard), Aug. 1998. Obsoleted by RFC 5925, updated by RFC 6691.

[8] J. Touch, A. Mankin, and R. Bonica, "The TCP Authentication Option." RFC 5925 (Proposed Standard), June 2010.

[9] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address Allocation for Private Internets." RFC 1918 (Best Current Practice), Feb. 1996. Updated by RFC 6761.

[10] M. Cotton, L. Vegoda, R. Bonica, and B. Haberman, "Special-Purpose IP Address Registries." RFC 6890 (Best Current Practice), Apr. 2013.

[11] IANA, "IPv4 Address Space Registry." <http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.txt>, mars 2013.

[12] A. Durand, R. Droms, J. Woodyatt, and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion." RFC 6333 (Proposed Standard), Aug. 2011.

[13] IANA, "IANA IPv6 Special Purpose Address Registry." <http://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.txt>, mai 2013.

[14] IANA, "Internet Protocol Version 6 Address Space." <http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.txt>, février 2013.

[15] IANA, "IPv6 Global Unicast Address Assignments." <http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.txt>, février 2013.

[16] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture." RFC 4291 (Draft Standard), Feb. 2006. Updated by RFCs 5952, 6052.

[17] R. Braden, "Requirements for Internet Hosts - Communication Layers." RFC 1122 (INTERNET STANDARD), Oct. 1989. Updated by RFCs 1349, 4379, 5884, 6093, 6298, 6633, 6864.

[18] S. Cheshire, B. Aboba, and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses." RFC 3927 (Proposed Standard), May 2005.

[19] S. Bradner and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices." RFC 2544 (Informational), Mar. 1999. Updated by RFCs 6201, 6815.

[20] J. Arkko, M. Cotton, and L. Vegoda, "IPv4 Address Blocks Reserved for Documentation." RFC 5737 (Informational), Jan. 2010.

[21] J. Weil, V. Kuarsingh, C. Donley, C. Liljenstolpe, and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space." RFC 6598 (Best Current Practice), Apr. 2012.

[22] M. Cotton, L. Vegoda, and D. Meyer, "IANA Guidelines for IPv4 Multicast Address Assignments." RFC 5771 (Best Current Practice), Mar. 2010.

[23] S. Deering, "Host extensions for IP multicasting." RFC 1112 (INTERNET STANDARD), Aug. 1989. Updated by RFC 2236.

[24] J. Mogul, "Broadcasting Internet Datagrams." RFC 919 (INTERNET STANDARD), Oct. 1984.

[25] N. Hilliard and D. Freedman, "A Discard Prefix for IPv6." RFC 6666 (Informational), Aug. 2012.

[26] R. Hinden, S. Deering, R. Fink, and T. Hain, "Initial IPv6 Sub-TLA ID Assignments." RFC 2928 (Informational), Sept. 2000.

[27] C. Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)." RFC 4380 (Proposed Standard), Feb. 2006. Updated by RFCs 5991, 6081.

[28] C. Popoviciu, A. Hamza, G. V. de Velde, and D. Dugatkin, "IPv6 Benchmarking Methodology for Network Interconnect Devices." RFC 5180 (Informational), May 2008.

[29] P. Nikander, J. Laganier, and F. Dupont, "An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers (ORCHID)." RFC 4843 (Experimental), Apr. 2007.

[30] G. Huston, A. Lord, and P. Smith, "IPv6 Address Prefix Reserved for Documentation." RFC 3849 (Informational), July 2004.

[31] B. Carpenter and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds." RFC 3056 (Proposed Standard), Feb. 2001.

[32] R. Hinden and B. Haberman, "Unique Local IPv6 Unicast Addresses." RFC 4193 (Proposed Standard), Oct. 2005.

[33] IANA, "Autonomous System (AS) Numbers." <http://www.iana.org/assignments/as-numbers/as-numbers.txt>, avril 2013.

[34] J. Mitchell, "Autonomous System (AS) Reservation for Private Use." RFC 6996 (Best Current Practice), July 2013.

[35] C. Systems, "Cisco IOS IP Routing: BGP Command Reference." <http://www.cisco.com/en/US/docs/ios/iproute_bgp/command/reference/irg_book.html>, mars 2011.

[36] "How the Internet in Australia went down under." BGPmon.net blog, 2012.

[37] Juniper Networks, "Technical Documentation - prefix-limit." <http://www.juniper.net/techpubs/en_US/junos11.4/topics/reference/configuration-statement/prefix-limit-edit-protocols-bgp.html>, octobre 2011.

[38] R. Gerhards, "The Syslog Protocol." RFC 5424 (Proposed Standard), Mar. 2009.

[39] S. Sangli, E. Chen, R. Fernando, J. Scudder, and Y. Rekhter, "Graceful Restart Mechanism for BGP." RFC 4724 (Proposed Standard), Jan. 2007.

[40] F. Baker and P. Savola, "Ingress Filtering for Multihomed Networks." RFC 3704 (Best Current Practice), Mar. 2004.

[41] T. Ylonen and C. Lonvick, "The Secure Shell (SSH) Protocol Architecture." RFC 4251 (Proposed Standard), Jan. 2006.

[42] D. Dugal, C. Pignataro, and R. Dunn, "Protecting the Router Control Plane." RFC 6192 (Informational), Mar. 2011.

[43] B. Carpenter, "RFC 1888 Is Obsolete." RFC 4048 (Informational), Apr. 2005. Updated by RFC 4548.

[44] C. Huitema and B. Carpenter, "Deprecating Site Local Addresses." RFC 3879 (Proposed Standard), Sept. 2004.

Acronyms

ANSSI	Agence nationale de la sécurité des systèmes d'information
ASIC	Application Specific Integrated Circuits
BGP	Border Gateway Protocol
EBGP	External Border Gateway Protocol
FIB	Forwarding Information Base
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IRRs	Internet Routing Registries
MAC	Message Authentication Code
RIB	Routing Information Base
RIR	Regional Internet Registry

About ANSSI

The French Network and Information Security Agency (ANSSI / Agence nationale de la sécurité des systèmes d'information) was created on 7th July 2009 as an agency with national jurisdiction ("service à compétence nationale").
By Decree No. 2009-834 of 7 July 2009 as amended by Decree No. 2011-170 of 11 February 2011, the agency has responsibility at the national level concerning the defence and security of information systems. It is attached to the Secretariat General for National Defence and Security (Secrétaire général de la défense et de la sécurité nationale) under the authority of the Prime Minister.
To learn more about ANSSI and its activities, please visit www.ssi.gouv.fr.

October 2014

Agence nationale de la sécurité des systèmes d'information
ANSSI - 51 boulevard de la Tour-Maubourg - 75700 PARIS 07 SP
Sites internet: `www.ssi.gouv.fr` et `www.securite-informatique.gouv.fr`
Messagerie: communication [at] ssi.gouv.fr