# BGP Secure Routing Extension

# – BGPSEC-IO –

*Version 0.2.0.6*

*User's Manual*

---


September 2018

www-x.antd.nist.gov/bgpsrx

bgpsrx-dev@nist.gov

Oliver Borchert, Kotikalapudi Sriram, Doug Montgomery

# Index

## Introduction

BGPSEC-IO (BIO) is developed by the Advanced Network Technology Division (ANTD) at the National Institute of Standards and Technology (NIST). This effort is supported by the Department of Homeland Security under the Secure Protocols for the Routing Infrastructure (SPRI) program and the NIST Information Technology Laboratory Cyber and Network Security Program.

BGPSEC-IO is a BGPsec traffic generator that allows to generate BGP UPDATE messages (RFC4271) including the BGP prefix origin validation state extended community (RFC 8097) as well as multi hop fully signed BGPsec UPDATE messages (RFC8205) and send them to a BGP / BGPsec capable router. In addition, it allows to only generate a multi hop fully signed BGPsec_PATH attribute to test plugins developed for the SRxCryptoAPI.
BIO allows to pre-generate traffic and store it as binary stream for later retrieval or generate traffic "on the fly" and this will reduce the generation overhead which is important for performance testing of BGPsec router implementations.

This traffic generator is capable of iBGP and eBGP configuration which also is handled differently within the BGPsec protocol. In case of missing keys which will lead to unsuccessful signed BGPsec paths, BIO allows to specify different fallback solutions:

(1) Generate simple BGP4 UPDATE
(2) Generate BGPsec UPDATE with "fake" signature using "fake" SKI's

The first solution is same as the operation a BGPsec speaker would do when talking to a non BGPsec speaker.
The second solution allows to test the correctness of BGPsec implementations by allowing the operator to pre-script the signature and SKI.

Furthermore, BIO allows to produce Wireshark-like outputs of send and received BGP UPDATES which can be filtered in various ways to only display information of interest.

## Installation and Requirements

Please see the SRxSoftwareSuite-5.0-QuickInstallGuide for the easy install. This version of BGPSEC-Io requires the SRxCryptoAPI 0.2.0

## Generating Keys:

Information on generating keys can be found in section "Key Generation Tools" of the SRxCryptoAPI (SCA) - User's Manual.
A set of example keys are provided with this distribution in the folder:

bgpsec-io/data

BIO uses the ski-list files to load the keys. A typical ski-list file has the following format:

```
<ASN>-SKI: <SKI – HEX STRING>

10-SKI: AB4D910F55CAE71A215EF3CAFE3ACC45B5EEC154
20-SKI: 47F23BF1AB2F8A9D26864EBBD8DF2711C74406EC
30-SKI: 3A7C104909B37C7177DF8F29C800C7C8E2B8101E
…
```

The SCA loading function splits the SKI in the following directory structure:

```
AB
 +--4D91
 |    +--0F55CAE71A215EF3CAFE3ACC45B5EEC154.cert
 |    +--0F55CAE71A215EF3CAFE3ACC45B5EEC154.der
47
 +--F23B
      +--F1AB2F8A9D26864EBBD8DF2711C74406EC.cert
      +--F1AB2F8A9D26864EBBD8DF2711C74406EC.der
…
```

BIO reads the ski list and pre-loads all keys into the memory. Then during the BGPsec_PATH generation it uses all pre-loaded keys to perform the signing.

## Scripting Updates:

The syntax for an update has the following format:

```
<PREFIX>[,<PATH>] with PATH=[ <ASN>[p<repetition>]]*[ ]*[I|V|N]?
Announced by BIO: "192.0.2.0/24"
BIO as transit:   "192.0.2.0/24, 65536 65536 65537"
BIO as transit:   "192.0.2.0/24, 65536p2 65537"
```

BIO will prepend its own AS number to the scripted update path, with the exception if the next hop has the same ASN as BIO.
The syntax of the path allows to specify a counter 'pCOUNT' to the AS. This means the ASN will be repeated multiple times. Following this format, Updates can either be handed over via command line, 'piped' in during program execution, or scripted within the configuration script. Additionally, the one time extension I, V, or N results in the generation of the extended community string containing the specified route origin validation. (I=invalid, V=valid, N=not found)

The configuration script contains two sections, where Updates can be scripted:

1. Within the session configuration
2. Outside the session configuration – global for all sessions*.

*Currently, only single sessions are supported.

```
Update = (
  "192.0.2.0/24",
  "192.0.2.0/24, 65536 65536 65537",
  "192.0.2.0/24, 65536p2 65537"
);
```

### Order Of Updates

BIO has many different forms on how to generate / provide updates.

1. Scripted in configuration file (session config)
2. Scripted in configuration file (global config)
3. Command line
4. Piped in via STDIO
5. Loaded from Binary file

And updates are read and processed in the same order as specified above.

**IMPORTANT**: SCA does not receive binary stored BGPsec UPDATES, only binary stored BGPsec_PATH attributes.

# Operational Modes

BIO can be operated in multiple modes:

1. BGP - Open BGP session to remote BGP router
2. CAPI - Load SRxCryptoAPI and perform validation requests
3. GEN-B - Generate BGPsec / BGP UPDATES and store them
4. GEN-C - Generate BGPsec_PATH attributes and store them

## BGP Mode

This mode allows to generate / play traffic to the remote peer by opening a BGP session to a remote server and playing all BGP / BGPsec traffic to the peer.
BIO receives all BGP messages from the peer but drops all BGP UPDATES and BGPsec UPDATES on arrival. Though, it is possible to have all BGP messages from the peer printed in Wireshark-like format.
In BGP mode all scripted updates, binary stored updates, and binary BGPsec_PATH attributes can be played.

## CAPI Mode

This mode allows to generate / play BGPsec_PATH attributes and call the SCA plugin to verify each attribute. This mode allows to

- Measure performance by timing each verification individually
- Testing the correctness and performance of SCA BGPsec crypto plugins

In BGP mode all scripted updates and binary BGPsec_PATH attributes can be used.

At the end of a CAPI run, BGPSEC-IO will display statistics on how many valid and invalid updates were processes and the average computation times. The measurements only include the time from starting to validate to getting the result. In other words, only the time consumed by SRxCryptoAPI is measured.

## GEN-B Mode

This mode allows to generate BGPsec and BGP updates for later replay in BGP mode

## GEN-C Mode

This mode allows to generate BGPsec_PATH attributes for later testing against CAPI plugins as well as replay in BGP mode.

## Signature Generation Modes

BGPSEC-IO allows to generate signatures in the following modes:

- CAPI : Let the SRxCryptoAPI generate the signature **(not available in this version)**
- BIO : Let BGPSEC-IO use its internal signature engine generate the signature.
- BIO-K1 : Same as BIO with the exception that the signature algorithm uses a valid pre-defined 'k' as specified in RFC 6969. Section A.2.5 example using "SHA-256 – sample"

```
k=A6E3C57DD01ABE90086538398355DD4C3B17AA873382B0F24D6129493D8AAD60
```

- BIO-K2 : Same as BIO with the exception that the signature algorithm uses a valid pre-defined 'k' as specified in RFC 6969. Section A.2.5 example using "SHA-256 – test"

```
k=D16B6AE827F17175E040871A1C7EC3500192C4C92677336EC2537ACAEE0008E0
```

## The Configuration File

To generate a BGPSEC-IO configuration file it is recommend to auto generate a fully functional configuration script.
To generate a configuration template call:

```
./bgpsecio -C <configuration-file>
```

This will generate a fully functional configuration file that can be customized.

## Configuration Settings

The following settings can be specified within a configuration file. All configuration file settings can be over-ruled by command line parameters. The configuration file has two sections, the global configuration section and the session configuration section(s). Currently BGPSEC-IO only supports a single session.

Also, it is recommended to configure the BGP peer in passive mode and have BGPSEC-IO initiating the session establishment.

## Global Configuration

### *Global - Crypto configuration*

**ski_file = <filename containing ASN-SKI key information>;**

This file contains all keys to be loaded (private / public). The format of the file content is "ASN-SKI: SKI-HEX-VALUE" as specified in *"Generating Keys"*.

**ski_key_loc = <path to the location keys are stored>;**

The path is the root folder where keys can be found. The storage of keys follows the structure described in *"Generating Keys"*.

**preload_eckey = true|false;**

Specifies if the OpenSSL EC_KEY should be generated during loading of the key. If not, the EC_KEY will be generated during the first usage of the key.

**capi_cfg = <configuration file>';**

Optional: Allows to specify a custom configuration to the SRxCryptoAPI (SCA). If not specified, the default configuration of SCA will be used.

### *Global - Operation configuration*

**mode = BGP|CAPI|GEN-B|GEN-C;**

Determines the operational mode of BGPSEC-IO. More information on the different modes can be found in the previous section *"Operational Modes"*.

**max = <integer>**

Maximum combined number of updates to process. Zero "0" for all.

**bin = <binary input file>;**

Optional: Specifies the file containing pre-generated BGPsec / BGP UPDATE messages and/or pre-generated BGPsec_PATH attributes.

**out = <binary input file>;**

Used only in *GEN-B* or *GEN-C* mode. This specifies the name of the file where the generated binary data will be stored in.

**appendOut = true|false;**

Specifies if the *binary out file* will be overwritten or extended if it already exists.

#### *Global - BGP / BGPsec configuration*

**only_extended_length = true|false;**

Allow to force the usage of two-byte length field within the BGPsec_PATH attribute regardless if the size of the attribute would allow a one byte length field. This includes the extended length flag being set to 1.

**updates = (<updates>*);**

Empty or a comma separated list of scripted updates. The syntax of scripted updates is explained in section *"Scripting Updates"*.

**session = ({Session Configuration}+)**

This contains one or more session configurations. Currently BGPSEC-IO only supports a single session configuration.
**IMPORTANT:** At least one session configuration must be provided.

### Session Configuration

The session configuration is required for all modes because it contains necessary information for the BGPSEC-IO - ASN as well as the peer ASN.

#### *Session - Local host configuration*

**asn = <as number>;**

The AS number BGPSEC-IO is using when connecting to the peer.

**bgp_ident = "<bgp identifier>";**

The BGP identifier BGPSEC-IO is using.

**hold_timer = <in seconds>;**

The hold timer for the BGP session.

**next_hop_ipv4 = "<IPv4 address of the next hop>";**

Optional: Specify the next hop value for IPv6 UPDATES. If not specified, the bgp_identifier is used.

**next_hop_ipv6 = "<IPv6 address of the next hop>";**

Optional: Specify the next hop value for IPv6 UPDATES. If not specified, the bgp_identifier is converted into a valid IPv6 address and used.

### *Session - Peer Configuration*

**peer_asn = <ASN of the peer>;**

The peer's AS number.

**peer_ip = "<IP address of the peer>";**

The peer IP address.

**peer_port = <0..65535>;**

The peer port address.

**disconnect = <time in seconds>;**

Allows to specify the session shutdown in seconds after the last BGPsec UPDATE or BGP UPDATE was send. If set to zero "0" BGPSEC-IO keeps the session open until the peer disconnects or CTRL+C is pressed.

### *Session - Extended Message draft support*

**ext_msg_cap = true|false;**

Specify if the extended message capability is used or not.

**ext_msg_liberal = true|false;**

Allow to enable/disable liberal behavior when receiving extended message capability.

**ext_msg_force = true|false;**

**Optional-Experimental:** Overwrite draft specification and force sending extended message regarding if negotiated or not.

### *Session - BGPsec Capabilities*

**bgpsec_v4_snd = true|false;**

Enable sending capability of IPv4 BGPsec UPDATES.

**bgpsec_v4_rcv = true|false;**

Enable receiving capability of IPv4 BGPsec UPDATES.

**bgpsec_v6_snd = true|false;**

Enable sending capability of IPv6 BGPsec UPDATES.

**bgpsec_v6_rcv = true|false;**

Enable receiving capability of IPv6 BGPsec UPDATES.

### *Session - Update Configuration*

Same as in global section

**updates = (<updates>*);**

Empty or a comma separated list of scripted updates. The syntax of scripted updates is explained in section *"Scripting Updates"*.

### *Session – Crypto Configuration*

**algo_id = <int>;**

The algorithm Identifier.

**signature_generation = CAPI|BIO|BIO-K1|BIO-K2**

Allows to specify the different signing modes. For more details, see the paragraph about *"Signature Generation Modes"*.

**null_signature_mode = DROP|FAKE|BGP4**

In case BGPSEC-IO (BIO) cannot generate a signature (missing or corrupted key) the signature generation will return a NULL value. This setting specifies how to behave in such an instance.

**DROP** : Ignore this update and keep going
**FAKE** : Use the signature scripted as *"fake_signature"*
**BGP4** : Don't generate a BGPsec UPDATE, use BGP4 UPDATE as fallback.

**fake_signature = "<HEX – VALUE – STRING>";**

Allows to pre-script a fake signature that will be used. This allows to test peer validation.

**fake_ski = "<40-character HEX SKI value>"**

Allows to pre-script a 40-character hex coded ski which will be used for fake signatures.

### *Session - Print Settings – Wireshark Format*

This section has two different types of printing BGP / BGPsec messages. One is on the sending side, one on the receiving side. Both configuration behave in the same manner. For this reason, the below documentation focuses on the sender side only.

**printOnSend = true|false|(detailed configuration);**

Allows to turn on or off printing of BGP messages send to peer. By just using true or false, all printing of all message types is either enabled or disabled. Using the detailed configuration allows cherry picking of update messages.

**printOnReceive = true|false|(detailed configuration)**

Same as "*printOnSend*".

#### Detailed Configuration settings of printOnSend and printOnReceive

**open = true|false;**
Print the BGP OPEN message.

**update = true|false;**
Print the BGP UPDATE message.

**keepalive = true|false;**
Print the BGP KEEPALIVE message.

**notification = true|false;**
Print the BGP NOTIFICATION message.

**unknown = true|false;**
Print the received message.

### *Session - Print Settings – BGPSEC-IO operations*

**printSimple = true|false**

This mode reduces the output form Wireshark format into a single line to reduce the output. This is helpful for printing large amounts of data. Prefix announcements and withdrawals are tagged with a "+" or "-" repectively.

**printPollLoop = true|false;**

Print BIO – peer socket polls – for debugging.

**printOnInvalid = true|false**

For CAPI mode only, print more information when an update validation returns invalid.

## Command Line Parameters

Command line parameters rule over scripted parameters. For instance, the operational mode is scripted as CAPI but the command line parameter specifies CAPI. The program will use the CAPI mode because the command line parameter overwrites the scripted mode.
The only difference are updates, they do not override, they are appended.
(see *"Scripting Updates"* in this document)

**-?, -h, -H, --help**
The help screen.

**-V, --version**
Display the version number.

**-f <config>, --config <config>**
config : the configuration file.

**-p <config>, --capi_cfg <config>**
config : an alternative SRxCryptoAPI configuration file.

**-u <prefix, path>, --update <prefix, path>**
prefix : prefix to be announced.
path : the list of AS numbers (right most is origin).

**-s <filename>, --ski_file <filename>**
Name of the SKI file generated by qsrx-publish

**-l <directory>, --ski_key_loc <directory>**
Specify the location where the keys and certificates are located.

**-m <type>, --mode <type>**
Enable the operational mode:
type BGP : run BGP player
type CAPI : run as SRxCryptoAPI tester.
type GEN : Generate the binary data.

**-a <asn>, --asn <asn>**
Specify the own AS number.

**-i <IPv4>, --bgp_ident <IPv4>**
The BGP identifier of the BGP daemon.

**-t <time>, --hold_timer <time>**
The hold timer in seconds (0 or >=3).

**-A <asn>, --peer_asn <asn>**
The peer as number.

**-I <IPv4>, --peer_ip <IPv4>**
The IP address of the peer.

**-P <port>, --peer_port <port>**
The port number of the peer.

**-M, --no_mpnlri**
DEPRECATED.
Disable MPNLRI encoding for IPv4 addresses. If disabled prefixes are encoded as NLRI only.

**-e, --no_ext_msg_cap**
Disable the usage of messages larger than 4096 bytes. This includes the capability exchange. (Default enabled)

**-L, --no_ext_msg_liberal**
Reject extended messages if not properly negotiated.

**--ext_msg_force**
Force sending extended messages regardless if capability is negotiated. Allows debugging the peer.

**-d <time>, --disconnect <time>**
The minimum time in seconds the session stays up after the last update was sent. The real disconnect time is somewhere between <time> and <holdTime> / 3.
A time of 0 "zero" disables the automatic disconnect.

**-E, --no_preload_eckey**
Disable pre-computation of EC_KEY structure during loading of the private and public keys.

**-b <filename>, --bin <filename>**
The filename containing the binary input data. Here only the first configured session will be used.

**-o <filename>, --out <filename>**
The filename where to write the output to - Here only the first configures session will be used.
Requires GEN mode!!

**-O, --appendOut**

If specified, the generated data will be appended to given outfile. In case the outfile does not exist, a new one will be generated.
Requires GEN mode!!

**-U, --max**

Allows to restrict the number of updates generated.

**-C <filename>**

Generate a configuration file. The configuration file uses the given setup (parameters, configuration file) or generates a sample file if no configuration is specified.

## Support

Before contacting us, please verify that the BGPSEC-IO check firewall settings. If nothing helps (not even a reboot), please contact us and we will try to help. In case of crashes, please provide a description on how to reproduce the crash and if possible a core dump.
To be informed of bug fixes or ask questions to the community, subscribe to the users email list by sending an email to bgpsrx-users-request@nist.gov with subscribe in the subject.

Questions to the developers and general contact information:

Email: bgpsrx-dev@nist.gov
Web: https://bgpsrx.antd.nist.gov

Developers:

Oliver Borchert oliver.borchert@nist.gov