TCP-AO
Configuration Guide

TCP-AO version 2019.1.0
23 August 2019

www.rtbrick.com

Headquarters

N-17L, 1st Floor, 18th Cross Rd
Sector 3, HSR Layout
Bengaluru, Karnataka
India 560102

+91 80 4850 5445

www.rtbrick.com

Support

support@rtbirck.com

Sales

+91 80 4850 5445

sales@rtbrick.com

Table of Contents

1 Introduction to TCP-AO

BGP, which connects routers with a TCP session, does not benefit directly from packet-level authentication. Authentication in BGP is a critical concern because BGP carries routing information all over the global public internet, creating a tempting target for route-status hackers, and, at the same time, making the authenticity of inter- and intra-AS routing information absolutely essential.

An initial attempt at TCP authentication in RFC 2385 added an MD5 signature to the TCP segment header. The TCP option Kind field value of 19 (decimal) indicated the presence of the MD5 signature in the TCP segment header. It did not take long before the restrictive nature of the length and type of the protection became apparent. Simply put, this method quickly became outdated.

RFC 5925 replaced the older method with a more flexible and improved authentication technique, using TCP header option Kind 29 (decimal) to indicate use of the TCP authentication option (TCP-AO).

According to RFC 5925, TCP-AO added:

- Stronger Message Authentication Codes (MACs)
- Better protection against replays for the long-lived TCP sessions common between routers
- A framework for the relationship between TCP authentication and security in general

The scope of the newer TCP-OA is indicated in the length of the RFCs: the 6 pages of RFC 2385 became 48 pages in RFC 5925. TCP-AO is compatible with the use of a Master Key Tuple (MKT), either static or with an out-of-band management technique. Both methods protect connections using the same MKT in repeated connection instances, as well as coordinating MKT changes between endpoints.

TCP-AO is compatible with the use of TCP MD5, but only with Kind = 29. However, if both Kind = 19 and Kind = 29 options appear in the same TCP header, or the header contains multiple Kind = 29 options, then the segment must be silently discarded.

The implications of TCP-AO use are considerable, such as the effects on TCP header size. For more information, see RFC 5925.

2 Implementation of TCP-AO

The RtBrick Full Stack (RBFS) supports three different MACs as described in RFC 5925:

- MD5 (RFC 2385 compatible)
- HMAC-SHA-1-96
- HMAC-SHA-256-128

Sections in this chapter include:

- TCP-AO Parameters
- TCP-AO Key Rollover

2.1 TCP-AO Parameters

TCP-AO for RBFS uses a set of parameters. The parameters are all mandatory and configured under an authentication_identifier that provides a convenient way to package the TCP-AO options.

The individual parameters gathered under an authentication_identifier are:

`local_ipv4_prefix`—A valid IPv4 prefix from which the connection is created.
`local_ipv6_prefix`—A valid IPv6 prefix from which the connection is created.
`remote_ipv4_prefix`—A valid IPv4 prefix to connect to.
`remote_ipv6_prefix`—A valid IPv6 prefix to connect to.
`local_port`—The local port that the connection uses.
`remote_port`—The remote port that the connection uses.
`authentication_type`—The authentication method used, either MD5, HMAC-SHA-1-96, or HMAC-SHA-256-128.
`send_key`—The secret key used by the sending side of the connection.
`send_key_id`—An identifier for the send key.
`receive_key1`—The primary key value expected from the other side of the connection.
`receive_key1_id`—An identifier for the primary expected key value.
`receive_key2`—The secondary key value expected from the other side of the connection.
`receive_key2_id`—An identifier for the secondary expected key value.

As noted above, all parameters are mandatory. Wild card support is provided by configuring a particular authentication_identifier on port 0. This configuration applies the same configuration to all ports in the system. The parameters are applied as local or remote depending on connection initiation.

The IPv4 and IPv6 prefix parameters provide an opportunity to control the range of prefixes onto which authentication is applied.

The key identifiers are used as defined in RFC 5925, although not supported in MD5. MD5 support is maintained for consistency, but the key identifiers are ignored if the authentication type is MD5. This allows the user to easily change the authentication type from MD5 to others on the fly, as needed.

The secret key string is stored in an AES encrypted format within the system. When entering this field, the user must prepend the plain text format with a 0 so that the system understands the input is in plain text. The string 0rtbrick is an example of this usage. In this case, the actual password is rtbrick and the prepended 0 indicates the plain text style is used.

The password string is stored in the system as
11e4946e31b406de98b3077aef03ed5a730293029329392320930293029302930

after encryption. This can be used as an input secret string for the remote system as well.

The command `show tcp authentication summary instance <instance name>` displays the encrypted information for the given instance.

2.2 TCP-AO Key Rollover

Normally, a rollover to a new key value disrupts the TCP session. For seamless key rollover, two receive keys and identifiers need to be configured.

Receive Side Authentication

Figure 1: TCP-AO Receive Side Authentication

Figure 1 shows how an incoming TCP message is processed at the receiver. First, the message digest from the TCP packet needs to be located and extracted. Then, the message digest is computed and validated against the first receive key.

If this initial step results in an __Auth-FAIL1__ event, then the message digest based on the *second* receive key is computed and validated. If there is another failure (an __Auth-FAIL2__ event), then the TCP-message is discarded and the TCP-failed-Auth counter is incremented.

.

3 TCP-AO Configuration Example

This section of the document details the configuration and use of the TCP-AO feature for the RBFS.

Sections in this chapter include:

- TCP-AO Topology
- TCP-AO Configuration

3.1 TCP-AO Topology

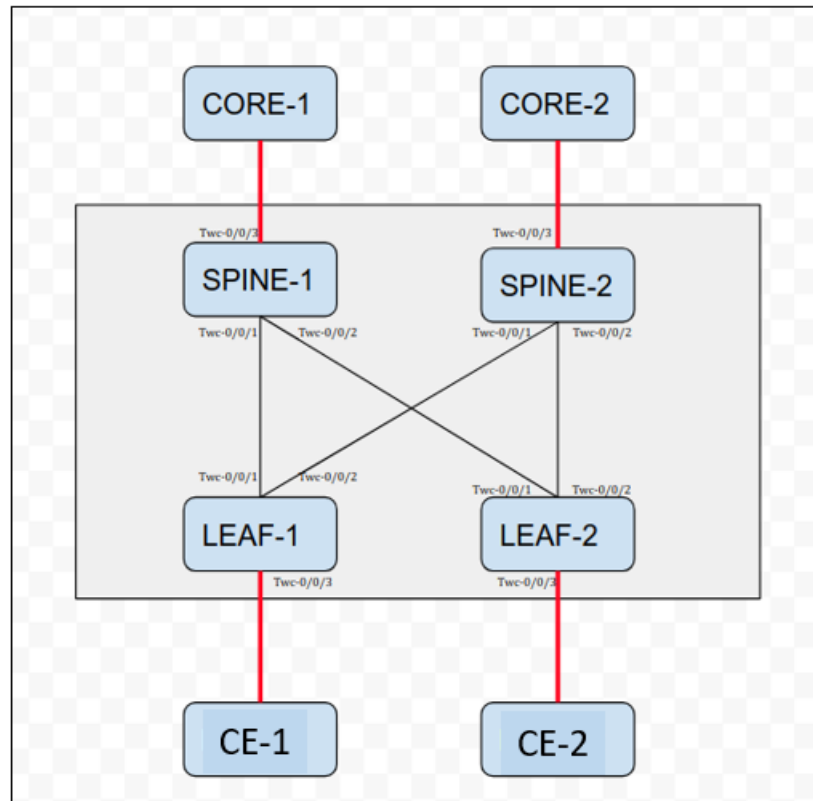The leaf/spine topology used in this example is shown in Figure 2.

Figure 2 The TCP-AO Example Topology.

In this topology:

- Leaf-1 and Leaf-2 are routers sitting close to the servers. These routers can be the equivalent of ToR (Top of Rack) boxes in the case of data center implementation, or Provider Edge (PE) routers in a typical ISP environment.
- Spine-1 and Spine-2 are the routers serving as route-reflectors, which distribute the routes across the leaf routers. They are connected to Core-1 and Core-2 routers (not included in this TCP-AO example). A BGP session runs between the Spine and Core routers.
- Customer Edge (CE-1 and CE-2) or related aggregation devices are routers that connect to the leaf devices. Typically, a BGP VRF session runs between a Leaf and CE device.

Only the leaf and spine devices are used for this example. All the configurations and outputs are shown for the same devices. All leaf and spine system are running the RBFS software.

3.2 TCP-AO Configuration

The code box below shows the example configuration to configure TCP session authentication for a BGP setup.

```json
{
 "objects": [
   {
     "attribute": {
       "authentication_identifier": "client",
       "receive_key2_id": 2,
       "receive_key2": "0rtbrick2",
       "receive_key1_id": 1,
       "local_ipv4_prefix": "192.168.10.1/16",
       "remote_ipv4_prefix": "192.168.10.2/16",
       "local_port": 0,
       "remote_port": 179,
       "authentication_type": "MD5",
       "send_key": "0rtbrick",
       "send_key_id": 1,
       "receive_key1": "11e4946e31b406de98b3077aef03ed5a7"
     }
   },
   {
     "attribute": {
       "authentication_identifier": "server",
       "receive_key2_id": 2,
       "receive_key2": "0rtbrick2",
       "receive_key1_id": 1,
       "local_ipv4_prefix": "192.168.10.1/16",
       "remote_ipv4_prefix": "192.168.10.2/16",
       "local_port": 179,
       "remote_port": 0,
       "authentication_type": "MD5",
       "send_key": "0rtbrick",
       "send_key_id": 1,
       "receive_key1": "11e4946e31b406de98b3077aef03ed5a7"
},
   }
 ],
  "table": {
   "table_name": "vrf_6.tcp.auth.config"
 }
}
```

Note that the code box above shows two JSON objects embedded in a single configuration file, one for the server and one for the client. The server BGP session uses local port number 179 and the remote

destination port can be any valid number, including 0. The port numbers for the client are the opposite: the local port is any valid number, including 0, and the remote port is 179. Both are required to match in this way for the session to activate.

Configurations can use the authentication types HMAC-SHA-1-96 or HMAC-SHA-256-128 as well.

The table name here is `vrf_6.tcp.auth.config`, indicating that this configuration applies to vrf_6.

In general, the table format to push the configuration to `<instance_name>.tcp.auth.config`. Please be sure to format the table name accordingly, because other forms are not supported.

4 TCP-AO Example Test Case

This section of the document shows how to bring up the authenticated BGP sessions and make sure that they are working properly.

Sections in this chapter include:

- Baseline BGP Session Without Authentication
- Configure TCP-AO Authentication
- Verify TCP-AO Configuration
- Verify Authenticated BGP Session Status
- Verify Authenticated TCP Connection Status

4.1 Baseline BGP Sessions Without Authentication

Initially, the example topology has established BGP sessions, but without authentication. The command `show bgp peer` shows the status of the BGP sessions (in this case, between a leaf and the spine routers).

```
root@bgp.iod.1.leaf1:/> show bgp peer
+----------------+----------------+----------------------+----------------------+-----------+-------------+--------------+
|     Peer       |    Domain      |       Peer IP        |      Source IP       | Remote-AS |    State    |  Interface   |
+----------------+----------------+----------------------+----------------------+-----------+-------------+--------------+
|   spine1@rtbrick.com | fe80::1:0:faff:febd:105 | fe80::1:0:faff:febd:501 |  65010*  | Established | twc-0/0/1/1/0 |
|   spine2@rtbrick.com | fe80::1:0:faff:febd:205 | fe80::1:0:faff:febd:502 |  65010*  | Established | twc-0/0/1/2/0 |
+----------------+----------------+----------------------+----------------------+-----------+-------------+--------------+
root@bgp.iod.1.leaf1:/>
[ brickbuilder ][                                                              (0*$bash)
```

For more information on configuring BGP on the RBFS software, see the **BGP Configuration Guide**.

4.2 Configure TCP-AO Authentication

The TCP-AO configuration for all device under test (DUT) boxes is shown below.

```
DUT: All boxes
    Source IPv6 Address: fe80::/16
    Remote IPv6 Address: fe80::/16
    Source port : 0
    Destination Port : 179
    Configuration parameters.
      Authentication Type : MD5
      Send Key           : 0rtbrick
      Receive Key1       :
11e4946e31b406de98b3077aef03ed5a7302930293293923209302932930293029
      Receive Key2       : 0rtbrick_alternate

    Source IPv6 Address: fe80::/16
    Remote IPv6 Address: fe80::/16
    Soucre port : 179
    Destination Port : 0
    Configuration parameters.
      Authentication Type : MD5
      Send Key           : 0rtbrick
      Receive Key1       :
11e4946e31b406de98b3077aef03ed5a7302930293293923209302932930293029
      Receive Key2       : 0rtbrick_alternate
```

4.3 Verify TCP-AO Configuration

The TCP-AO configuration is displayed with the `show tcp authentication summary` command.

Here is the result for `leaf1`:

```
root@confd.leaf1:/> show tcp authentication summary
Authentication Identifier: client, Local IPv6 Prefix: ::/0, Remote IPv6 Prefix: fe80::/16
  Source Port        : 0
  Destination Port   : 179
  Authentication Type : HMAC-SHA-256-128
  Send Key           : 11e4946e31b406de98b3077aef03ed5a7
  Receive Key1       : 11e4946e31b406de98b3077aef03ed5a7
  Receive Key2       : 1e7bf7ba323e5f21be736616c8657f786
Authentication Identifier: server, Local IPv6 Prefix: ::/0, Remote IPv6 Prefix: fe80::/16
  Source Port        : 179
  Destination Port   : 0
  Authentication Type : HMAC-SHA-256-128
  Send Key           : 11e4946e31b406de98b3077aef03ed5a7
  Receive Key1       : 11e4946e31b406de98b3077aef03ed5a7
  Receive Key2       : 1e7bf7ba323e5f21be736616c8657f786
```

Here is the result for spine1:

```
root@confd.spine1:/> show tcp authentication summary
Authenticaion Identifier: to_leaf1_md5_v6, Local IPv6 Prefix: fe80::1:0:faff:febd:105/128, Remote IPv6 Prefix: fe80::1:0:faff:febd:501/128
  Source Port        : 0
  Destination Port   : 0
  Authentication Type : MD5
  Send Key           : 11e4946e31b406de98b3077aef03ed5a73029302932939232093029329302930229
  Receive Key1       : 11e4946e31b406de98b3077aef03ed5a73029302932939232093029329302930229
  Receive Key2       : 1f2b2400da81d1827ebad7ab4bf74ba9d8aa08d0b0e912e73709e3116222e4c95
Authentication Identifier: to_leaf1_md5_v6_fake, Local IPv6 Prefix: fe80::1:0:faff:febd:105/128, Remote IPv6 Prefix: fe80::1:0:faff:febd:501/128
  Source Port        : 200
  Destination Port   : 200
  Authentication Type : MD5
  Send Key           : 11e4946e31b406de98b3077aef03ed5a73029302932939232093029329302930229
  Receive Key1       : 11e4946e31b406de98b3077aef03ed5a73029302932939232093029329302930229
  Receive Key2       : 1f2b2400da81d1827ebad7ab4bf74ba9d8aa08d0b0e912e73709e3116222e4c95
root@confd.spine1:/>
[ brickbuilder ][                                                        0$ bash  1*$bash   2-$ bash
```

Verify that all parameters are correct.

4.4 Verify Authenticated BGP Session Status

After TCP-AO configuration and application, the BGP sessions should still be up and established. BGP session status is displayed with the `show bgp peer` command.

```
root@bgp.iod.1.leaf1:/> show bgp peer
+----------------+----------------+-------------------+-------------------+-----------+--------------+---------------+
|     Peer       |    Domain      |      Peer IP      |    Source IP      | Remote-AS |    State     |   Interface   |
+----------------+----------------+-------------------+-------------------+-----------+--------------+---------------+
|   spine1@rtbrick.com  | fe80::1:0:faff:febd:105 | fe80::1:0:faff:febd:501 |  65010*   | Established  | twc-0/0/1/1/0 |
|   spine2@rtbrick.com  | fe80::1:0:faff:febd:205 | fe80::1:0:faff:febd:502 |  65010*   | Established  | twc-0/0/1/2/0 |
+----------------+----------------+-------------------+-------------------+-----------+--------------+---------------+
root@bgp.iod.1.leaf1:/> ~~
[ brickbuilder ][                                                        (0*$bash)  1-$ bash  2$ bash
```

The same status should be displayed for the other leaf and spine devices.

4.5 Verify Authenticated TCP Status

After TCP-AO configuration and application, the TCP connections should display authentication flags and options. TCP connection and session status is displayed with the show `tcp status detail` command.

```
Local IPv6 Address: fe80::1:0:faff:febd:105, Remote IPv6 Address: fe80::1:0:faff:febd:501, Local Port: 179, Remote Port: 49172
  State               : Established
  Internal
    Options           : -- | Keepalive | --
    TOS               : 0
    TTL               : 255
    Priority          : 1
    Flags             : -|-|-|-|-|-|-|Wnd Scale|MD5 Auth|-|-
    Last Trigger      : 34
    Timer             : 21025
  Timers
    Poll              : 0
    Poll Interval     : 0
    Retransmission    : 65535
  Receiver
    Expected Sequence    : 1995447
    Available Window     : 65536
    Announced Window     : 64814
    Annoucned Wnd Rt Edge : 2060261
    MSS                  : 1440
    RTT Estimate         : 0
  Timeout
    Sequence          : 2061812
    Retransmission    : 8
    Retransmissions   : 0
    Duplicate Acks    : 0
    Highest Ack'd Sequence: 2061831
  Congestion
    Window            : 33017
    Persist Count     : 0
    Persist Backoff   : 0
    Send Scale        : 5
    Receive Scale     : 5
  Sender
    Next Seq to send     : 2061831
    Last Wnd update Seq   : 1995428
    Last Wnd update Ack   : 2061831
    Window               : 64864
    Max Window Announced  : 2097120
    Acknowledged         : 0
    Send Buf             : 57344
    Send Queue length    : 0
    Unsent Oversize      : 0
    TS Last ack sent     : 0
  Keepalive
    Next Keepalive Idle  : 7200000
    Keepalive Interval   : 75000
    Keepalive Count      : 9
    Keep Sent Count      : 0
  Authentication
    Auth Type         : MD5
    Send Key          : 11e4946e31b406de98b3077aef03ed5a73029302932939232093029329302930229
    Recv Key1         : 11e4946e31b406de98b3077aef03ed5a73029302932939232093029329302930229
    Recv Key2         : 1f2b2400da81d1827ebad7ab4bf74ba9d8aa08d0b0e912e73709e3116222e4c95
    Latest Sent Digest    : 501f67f0229ecf6348677bb83f9b035a
    Latest Received Digest: 969b7b46839d74c9aeef2342a4355395
Local IPv6 Address: fe80::1:0:faff:febd:106, Remote IPv6 Address: fe80::1:0:faff:febd:601, Local Port: 49154, Remote Port: 179
  State               : Established
[ brickbuilder ][                                                      0-$ bash  (1*$bash)  2$ bash
```

The same status should be displayed for the other leaf and spine devices.

5 Stressing the TCP-AO Example

This section of the document shows how configuration changes can force TCP-AO to take certain actions.

Sections in this chapter include:

- Force Use of Receive Key2
- Force Key Mismatch
- Change Authentication Type

5.1 Force Use of Receive Key2

Up to this point, the example topology has established BGP sessions using TCP-AO. The command show bgp peer shows the status of the BGP sessions. If the configuration send key changes, but still matches the second receive key, the connection re-establishes itself, and does not flap.

To demonstrate this, change the configuration on `Spine1` to use `0rtbrick_alternate` as the send key. This change is shown in bold.

```
DUT: Spine1
    Source IPv6 Address: fe80::/16
    Remote IPv6 Address: fe80::/16
    Source port : 0
    Destination Port : 179
    Configuration parameters.
      Authentication Type : MD5
      Send Key             : 0rtbrick_alternate
      Receive Key1         :
11e4946e31b406de98b3077aef03ed5a730293029329392320930293293029302930293029
      Receive Key2         : 0rtbrick_alternate
```

This new configuration forces the connection to re-establish with the alternate key, which is shown using the `show tcp authentication summary` command on `Spine1`.

The `authentication-secret-mismatch` counter should be zero. This can be shown by dumping the contents of the `global.tcp.auth.statistics`, as shown below.

5.2 Force Key Mismatch

If the configuration send key changes to a key that the receiver does not recognize at all, the connection will flap and not re-establish itself.

To demonstrate this, change the secret key on a system such as Spine1 to send a key not configured on another system. `Spine1` should log `authentication-secret-mismatches`, so the counter should be greater than 0.

To revert to the established session state, change the send key configuration back to one of the two valid send keys. The authentication-secret-mismatches counter should stabilize and stop incrementing.

5.3 Change Authentication Type

If the configuration Authentication Type changes from MD5 to SHA-1 or SHA-2, the sessions do not log `authentication-secret-mismatches`, but log `authentication-algo-mismatches`.

In other words:
- The `authentication-secret-mismatches` counter remains stable or 0
- The `authentication-algo-mismatches` counter increments

Changing the configuration back to matching authentication algorithms re-establishes the session and stabilizes the counters.

6 Deleting TCP-AO

This section of the document shows what happens when TCP-AO is deleted.

If TCP-AO is removed, the session should flap and not be re-established until both end systems of the connection match configurations again.

Check the session uptime to make sure the session has falled.

The assumption for lossless behavior is that the TCP stack silently discards (ignores) the following:
- Unauthenticated segments
- Authenticated segments when no authentication is configured

Both sides will retransmit until both ends of the session are configured with matching authentication configurations.

7 Troubleshooting TCP-AO

This section of the document shows how to address TCP-AO issues. There are a couple of things to check if TCP-AO results are not what is expected.

- Most TCP-AO configuration-related issues can be caught through logging. To enable the logging, use the following command:

  set tcp log-level 5

- Log levels can be from 1 to 5, which have increasing order of verbosity. You can display the log messages stored in /var/log/upstart/<app_name>.log

- Make sure that TCP-AO configurations are applied on the sessions of interest. Check that by displaying the TCP sessions and making sure that all options are set. Display the TCP status detail with the `show tcp status detail` command.

Make sure to apply the configurations for both server and client end for protocol sessions such as BGP.

Let us know if you find any other useful tips!