CONFD RESTCONF TOKEN AUTHENTICATION

tail-f a Cisco company

Table of Contents

1. Abstract

The RESTCONF standard as defined in RFC8040 specifies how to map YANG data models to a RESTful interface. ConfD has supported RESTCONF since it was first defined and automatically renders a northbound RESTCONF API from the YANG data models.

Over time, ConfD's RESTCONF implementation has gained new features such as the support which was added to generate Swagger definitions from YANG data models.

Another recent new feature is support for RESTCONF token authentication as an alternative to using username/password authentication. RESTCONF token authentication was added to ConfD Premium starting with version 6.6.

This application note examines what RESTCONF token authentication is, how it works, and what you need to do in order to be able to make use of this feature.

2. Background

Each RESTCONF API call has to be authenticated. This can be done in the traditional way by passing a user name and password. For example, see following curl command.

```
curl -v -is \
        -u admin:admin \
        -H "Accept: application/yang-data+xml" \
        http://localhost:8008/restconf/data/restconf-state
```

Some RESTful servers support token based authentication in which a token is often passed using X-Auth-Token header. Here is a corresponding curl command using the token rctoken.

```
curl -v -is \
        -H "X-Auth-Token: rctoken" \
        -H "Accept: application/yang-data+xml" \
        http://localhost:8008/restconf/data/restconf-state
```

The token is mapped to specific user. It can be fixed (i.e. always the same) or it can be changed. The changed token can be returned in the response to original request and then can be used in subsequent RESTCONF requests.

There are PROs and CONs to using RESTful token authentication.

PROs:

- It is not necessary to send the username and password in each RESTCONF request.
- Better security, mainly when the token is not fixed and changes, the token can expire.

CONs:

- When the token changes, one can view token authentication as a possible violation of the RESTful concept, as the token can be seen as state.

3. RESTCONF Token Authentication in ConfD

This application note will describe:

- How to setup ConfD to accept a specific RESTCONF authentication token which is passed in the HTTP(S) header (X-Auth-Token), with the use of /aaa/externalValidation in confd.conf.
- How to use the RESTCONF authentication token in RESTCONF requests by passing the token in X-Auth-Token.
- How to use normal (username/password) authentication to get an initial RESTCONF token from ConfD which is returned in X-Auth-Token with the use of /aaa/restconf/tokenResponse in confd.conf.
- How to make a token temporary and change it with each RESTCONF request. The new token is returned in X-Auth-Token in the response.

For demonstration purposes, we will use the RESTCONF example distributed with ConfD which is found in the examples.confd/restconf/yang-patch directory of a ConfD Premium installation. It is recommended to copy this example to some other location because it will be modified to show how to use RESTCONF token authentication.

3.1 Setup RESTCONF Token Authentication in ConfD

To use an authentication token with RESTCONF, we need to set-up AAA external validation.

We can also instruct ConfD to return the token in each request.

If we want to get the token after initial username/password authentication, we need to also set-up AAA external authentication.

The following subsections will show how to do these steps in our RESTCONF example based on examples.confd/restconf/yang-patch.

3.2 AAA External Validation

As you would expect from the name, AAA external validation is part of ConfD AAA functionality. It is used to perform the validation of the passed token. Currently , AAA external authentication can only be used with RESTCONF.

For more information information about this feature, please, see the chapter AAA Infrastructure in the ConfD User Guide.

NOTE:

The term AAA external validation should not be confused with external validation of configuration. The AAA external validation validates the authentication token being passed in a RESTCONF request, whereas configuration external validation validates configuration being applied inside transaction and uses a different mechanism.

3.2.1 Usage Example

To set-up AAA external validation, edit the AAA section of the ConfD configuration file (confd.conf) and add the externalValidation element. In addition to enabling this, a path pointing to an executable program or script which performs the validation has to be provided.

```
<aaa>
        ...
        <externalValidation>
        <enabled>true</enabled>
        <executable>./ext-val.sh</executable>
        </externalValidation>
        ...
</aaa>
```

The executable reads from stdin the token which is passed by ConfD from the received RESTCONF request in the form “[token;]\n” and then writes to stdout user information associated with the token in the form “accept $groups $uid $gid $supplementary_gids $HOME $USER\n”.

Where:

- $groups is a space separated list of the group names the user is a member of.
- $uid is the UNIX integer user id ConfD should use as default when executing commands for this user.
- $gid is the UNIX integer group id ConfD should use as default when executing commands for this user.
- $supplementary_gids is a space separated list of additional UNIX group ids the user is also a member of, can be empty
- $HOME is the directory which should be used as HOME for this user when ConfD executes commands on behalf of this user.
- $USER is the user derived from mapping the token.

./ext-val.sh:

```
#!/bin/sh

read INPUT
token=$(echo $INPUT | sed -n 's/\[\(.*\);\]/\1/p')

echo $INPUT $token >/tmp/token.txt

if [ "$token" = "rctoken" ]; then
        echo accepted >> /tmp/token.txt
        echo "accept 9000 20 '/tmp' admin"
else
        echo "reject 'token validation failure'"
fi
```

NOTE:
As an executable, we are using a simple shell script with a hardcoded token (the token is rctoken). In a real scenario, the executable will usually be a binary application which accesses some user/authentication system which generates the tokens.

We have enabled AAA external validation and provided an executable (shell script) that performs token validation and associates the token with the admin user.

To test this, we will use sources from examples.confd/restconf/yang-patch. First, modify confd.conf by adding the externalValidation element (from above) to the aaa element. Next, copy (or create) the ext-val.sh shell script in the example directory and make sure that it is executable - chmod a+x ./ext-val.sh.

Now you can rebuild and start the example:

```
make clean all start
```

In another terminal window, send a RESTCONF message with the token:

```
curl -v -H "X-Auth-Token: rctoken" \
        -H "Accept: application/yang-data+xml" \
        http://localhost:8008/restconf/data/restconf-state
```

The response shows that the RESTCONF request is accepted:

```
< HTTP/1.1 200 OK
< Server:
< Date: Tue, 13 Nov 2018 13:59:44 GMT
< Last-Modified: Fri, 01 Jan 1971 00:00:00 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1542-117340-647237
< Content-Type: application/yang-data+xml
< Transfer-Encoding: chunked
< Pragma: no-cache
<

<restconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf-monitoring"
                xmlns:rcmon="urn:ietf:params:xml:ns:yang:ietf-restconf-monitor-
ing">
<capabilities>
  <capability>urn:ietf:params:restconf:capability:defaults:1.0?basic-mode=expli
cit</capability>
  ...
</capabilities>
</restconf-state>
```

Now, disable external validation in confd.conf:

```
 <aaa>
    ...
    <externalValidation>
      <enabled>false</enabled>
      <executable>./ext-val.sh</executable>
    </externalValidation>
    ...
  </aaa>
```

Send another RESTCONF message:

```
make start
curl -v -H "X-Auth-Token: rctoken" \
     -H "Accept: application/yang-data+xml" \
     http://localhost:8008/restconf/data/restconf-state
```

Now, the response shows that the RESTCONF request is rejected:

```
< HTTP/1.1 401 Unauthorized
< Server:
< Date: Tue, 13 Nov 2018 14:04:09 GMT
< Content-Length: 169
< Content-Type: application/yang-data+xml
< WWW-Authenticate: Basic realm="restconf"
< Vary: Accept-Encoding
<
<errors xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
  <error>
    <error-tag>access-denied</error-tag>
    <error-type>protocol</error-type>
  </error>
</errors>
```

Now, try authentication with username/password:

```
curl -v -u admin:admin \
     -H "Accept: application/yang-data+xml" \
     http://localhost:8008/restconf/data/restconf-state
```

The RESTCONF request is accepted and shows the correct response as when a token was used:

```
< HTTP/1.1 200 OK
< Server:
< Date: Tue, 13 Nov 2018 14:06:07 GMT
< Last-Modified: Fri, 01 Jan 1971 00:00:00 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1542-117841-443909
< Content-Type: application/yang-data+xml
< Transfer-Encoding: chunked
< Pragma: no-cache
<

<restconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf-monitoring"
                xmlns:rcmon="urn:ietf:params:xml:ns:yang:ietf-restconf-monitor-
ing">
<capabilities>
  <capability>urn:ietf:params:restconf:capability:defaults:1.0?basic-mode=explic
it</capability>
  ...
</capabilities>
</restconf-state>
```

Enable AAA external validation in confd.conf again and restart ConfD via make start for the next section.

```
<aaa>
    ...
    <externalValidation>
      <enabled>true</enabled>
      <executable>./ext-val.sh</executable>
    </externalValidation>
    ...
  </aaa>
```

3.3 Getting an Authentication Token from ConfD with Token Authentication

It is possible to instruct ConfD to return an authentication token in the response.

To do this, we first have to add the tokenResponse element to RESTCONF section of the ConfD configuration file confd.conf.

confd.conf

```
<restconf>
    ...
    <tokenResponse>
      <xAuthToken>true</xAuthToken>
    </tokenResponse>
</restconf>
```

We also have to modify the AAA external validation executable to return accept_token instead of accept.

“accept_token $groups $uid $gid $supplementary_gids $HOME $USER $token\n”

- $token is an arbitrary string, returned as token in the response

./ext-val.sh:

```
#!/bin/sh

read INPUT
token=$(echo $INPUT | sed -n 's/\[\(.*\);\]/\1/p')

echo $INPUT $token >/tmp/token.txt

if [ "$token" = "rctoken" ]; then
    echo accepted >> /tmp/token.txt
    echo "accept _ token 9000 20 '/tmp' admin rctoken"
else
    echo "reject 'token validation failure'"
fi
```

Restart ConfD:

```
make start
```

Send a RESTCONF request:

```
curl -v -H "X-Auth-Token: rctoken" \
     -H "Accept: application/yang-data+xml" \
     http://localhost:8008/restconf/data/restconf-state
```

Now, you can see the token is part of returned response header in the field X-Auth-Token:

```
< HTTP/1.1 200 OK
< Server:
< Date: Tue, 13 Nov 2018 14:45:40 GMT
< Last-Modified: Fri, 01 Jan 1971 00:00:00 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1542-117841-443909
< Content-Type: application/yang-data+xml
< Transfer-Encoding: chunked
< X-Auth-Token: rctoken
< Pragma: no-cache
<

<restconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf-monitoring"
                xmlns:rcmon="urn:ietf:params:xml:ns:yang:ietf-restconf-monitor-
ing">
<capabilities>
  <capability>urn:ietf:params:restconf:capability:defaults:1.0?basic-mode=expli
cit</capability>
  ...
</capabilities>
</restconf-state>
```

You can modify the token in the accept_token response, restart ConfD, and see that the returned token is the modified one and not the one passed with curl command. This enables the implementation of a temporary token which is changed for each request.

3.4 Getting an Authentication Token from ConfD with Username/Password Authentication

If you invoke a RESTCONF request with username/password authentication, the token is not returned in the response.

```
curl -v -u admin:admin \
     -H "Accept: application/yang-data+xml" \
     http://localhost:8008/restconf/data/restconf-state
```

The reason for this is that with username/password authentication the AAA external validation is not used.

This can represent a problem if we do not know the token and we want to get it from ConfD to use it later on. Luckily, ConfD provides a solution to this with AAA External authentication (see the ConfD User Guide, chapter The AAA infrastructure).

Similar to AAA external validation, we need to update confd.conf in order to enable AAA external authentication as well as to configure an executable for handling it. For the purposes of demonstration, we will use a simple bash script. We also need to change the authentication order, so that AAA external authentication takes place before Local authentication and PAM.

NOTE:
The authentication order has to be changed only if user can be authenticated by ConfD locally or by PAM.

The executable reads from stdin the username and password passed from ConfD in the form "[user;password;]\n" and writes to stdout user information in the form "accept_token $groups $uid $gid $supplementary_gids $HOME $token\n".

The individual parameters are the same as for AAA external validation.

confd.conf:

```
<aaa>
  ...
  <externalAuthentication>
    <enabled>true</enabled>
    <executable>./ext-auth.sh</executable>
  </externalAuthentication>
  <authOrder>externalAuthentication localAuthentication pam</authOrder>
  ...
</aaa>
```

ext-auth.sh:

```
#!/bin/sh

read INPUT
user=$(echo $INPUT | sed -n 's/\[\(.*\);\(.*\);\]/\1/p')
pass=$(echo $INPUT | sed -n 's/\[\(.*\);\(.*\);\]/\2/p')

if [ "$user" = "admin" ] && [ "$pass" = "admin" ]; then
    echo "accept _ token 9000 20 '/tmp' rctoken"
else
    echo "reject 'permission denied'"
fi
```

To test this, enable externalAuthentication in confd.conf and create/copy ext-auth.sh to the example directory and make it executable with chmod +x ext-auth.sh.

Restart ConfD:

```
make start
```

Invoke a RESTCONF request with username/password authentication:

```
curl -v -u admin:admin \
     -H "Accept: application/yang-data+xml" \
     http://localhost:8008/restconf/data/restconf-state
```

Now, you should see the token returned from the ext-auth.sh in the response header in the X-Auth-Token field:

```
< HTTP/1.1 200 OK
< Server:
< Date: Wed, 14 Nov 2018 10:07:49 GMT
< Last-Modified: Fri, 01 Jan 1971 00:00:00 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1542-189651-525013
< Content-Type: application/yang-data+xml
< Transfer-Encoding: chunked
< X-Auth-Token: rctoken
< Pragma: no-cache
<

<restconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf-monitoring"
                xmlns:rcmon="urn:ietf:params:xml:ns:yang:ietf-restconf-monitor-
ing">
<capabilities>
  <capability>urn:ietf:params:restconf:capability:defaults:1.0?basic-mode=explic
it</capability>
  ...
</capabilities>
</restconf-state>
```

3.5 Temporary Token

At this point, we have all the knowledge needed to implement a token that is valid only temporarily and changes over time.

For this, we only need to return changed token in the response (in ext-val.sh and ext-auth.sh) and make sure it is accepted with in the next RESTCONF call (in ext-val.sh).

As a simple test, we can use a text editor and modify the token in the executable scripts.

4. Conclusion

We have seen RESTCONF token authentication is, how ConfD supports it, and what you need to do to make use of it. This support is implemented with use of AAA external validation and AAA external authentication and requires writing validation and authentication executables. When a token should be in a response to a username/password request, the AAA local authentication has to be suppressed.

The executables can be simple shell scripts, but the production applications should be more sophisticated and, for security reasons, implemented as a binary. Applications should also take into consideration that multiple clients can invoke RESTCONF messages and they should identify clients in subsequent calls by doing something such as making a client ID be part of the returned token.

With the information which has been presented in this application note, you are now ready to make use of this RESTCONF authentication option.

5. For More Information

For more information about ConfD, visit http://www.tail-f.com

Whitepaper: “Inside RESTCONF”: http://info.tail-f.com/inside-restconf

For more information about RESTCONF, visit https://tools.ietf.org/html/rfc8040

ConfD User Guide, chapter “The AAA Infrastructure”

tail-f a Cisco company

www.tail-f.com
info@tail-f.com

Corporate Headquarters

Sveavagen 25
111 34 Stockholm
Sweden
+46 8 21 37 40