



# Route Origin Validation Lab

## Part-1: Installing FORT Validator

---

In Part 1, you will install an RPKI Validator. There are a few validators available, and in this section, there are steps for installing the Fort Validator. For more detail about installing other validators refer to the APNIC blog - <https://blog.apnic.net/2019/10/28/how-to-installing-an-rpki-validator/>

FORT is a routing security initiative by LACNIC and NIC.MX, for more information refer to <https://fortproject.net/en/home>

The steps to complete this lab are: \* SSH to the server \* Update software \* Install required software \* Install Fort validator \* Update RPKI cache

### Login Details

- Username `apnic` and password `training` .

### Preinstalled packages

To save time, the following essential package(s) have been preinstalled on the containers:

- `rsync`

### Lab Setup

For this lab, we will use [FORT](#) from NIC México as the RPKI validator.

1. Login to your server (SSH from the jumhost to your container using the `username` and `password` given above), where `x` is your VM number:

```
ssh apnic@192.168.30.x
```

NOTE: Type `yes` if asked about wanting to continue connecting

Password = `training`

## 2. Update the repository

```
sudo apt update && sudo apt dist-upgrade
```

Password = `training`

◦ **NOTE:** Please note that FORT has the following dependencies:

- [jansson](#)
- libcrypto ([LibreSSL](#) or [OpenSSL](#) >=1.1)
- rsync

◦ To install the dependencies:

```
sudo apt-get install -y nano autoconf automake build-essential libjansson-dev  
libssl-dev pkg-config rsync libcurl4-openssl-dev libxml2-dev
```

◦ Verify the OpenSSL version

```
openssl version -a
```

```
tashi@fort:~$ openssl version -a  
OpenSSL 1.1.1 11 Sep 2018
```

3. Fetch the source file from the Github repository. Version 1.5.0 was the latest at the time of writing this lab. Please visit the website to confirm the latest version <https://github.com/NICMx/FORT-validator/releases/>

```
cd ~  
wget https://github.com/NICMx/FORT-validator/releases/download/v1.5.0/fort-1.5.0.  
tar.gz
```

```
apnic@fort: ~
File Edit View Search Terminal Help
de
Setting up build-essential (12.8ubuntu1.1) ...
Setting up libalgorithm-diff-xs-perl (0.04-6) ...
Setting up libalgorithm-merge-perl (0.08-3) ...
Processing triggers for systemd (245.4-4ubuntu3.11) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
apnic@fort:~$ cd ~
apnic@fort:~$ wget https://github.com/NICMx/FORT-validator/releases/download/v1.5.0/fort-1.5.0.tar.gz
--2021-08-31 04:01:53-- https://github.com/NICMx/FORT-validator/releases/download/v1.5.0/fort-1.5.0.tar.gz
Resolving github.com (github.com)... 192.30.255.113
Connecting to github.com (github.com)|192.30.255.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-releases.githubusercontent.com/147424126/3e56a980-6b98-11eb-8873-2821d150ba43?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20210831%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20210831T040153Z&X-Amz-Expires=300&X-Amz-Signature=6a77369721ad6bc3cc122bf1c224ee604b13966573db68449d0e6aecefeb6c8f&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=147424126&response-content-disposition=attachment%3B%20filename%3Dfort-1.5.0.tar.gz&response-content-type=application%2Foctet-stream [following]
--2021-08-31 04:01:53-- https://github-releases.githubusercontent.com/147424126/3e56a980-6b98-11eb-8873-2821d150ba43?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20210831%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20210831T040153Z&X-Amz-Expires=300&X-Amz-Signature=6a77369721ad6bc3cc122bf1c224ee604b13966573db68449d0e6aecefeb6c8f&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=147424126&response-content-disposition=attachment%3B%20filename%3Dfort-1.5.0.tar.gz&response-content-type=application%2Foctet-stream
Resolving github-releases.githubusercontent.com (github-releases.githubusercontent.com)... 185.199.111.154, 185.199.110.154, 185.199.109.154, ...
Connecting to github-releases.githubusercontent.com (github-releases.githubusercontent.com)|185.199.111.154|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 560200 (547K) [application/octet-stream]
Saving to: 'fort-1.5.0.tar.gz'

fort-1.5.0.tar.gz  100%[=====>] 547.07K  --.-KB/s   in 0.03s

2021-08-31 04:01:53 (15.5 MB/s) - 'fort-1.5.0.tar.gz' saved [560200/560200]

apnic@fort:~$
```

Extract the files from the tar.gz archive

```
tar zxvf fort-1.5.0.tar.gz
```

```
apnic@fort: ~
File Edit View Search Terminal Help
fort-1.5.0.tar.gz 100%[=====>] 547.07K --.-KB/s in 0.03s
2021-08-31 04:01:53 (15.5 MB/s) - 'fort-1.5.0.tar.gz' saved [560200/560200]

apnic@fort:~$
apnic@fort:~$ ls
Downloads fort-1.5.0.tar.gz
apnic@fort:~$ tar zxvf fort-1.5.0.tar.gz
fort-1.5.0/
fort-1.5.0/Makefile.am
fort-1.5.0/examples/
fort-1.5.0/examples/config.json
fort-1.5.0/examples/demo.slurm
fort-1.5.0/examples/tal/
fort-1.5.0/examples/tal/afrinic.tal
fort-1.5.0/examples/tal/ripe.tal
fort-1.5.0/examples/tal/apnic.tal
fort-1.5.0/examples/tal/lacnic.tal
fort-1.5.0/NOTICE
fort-1.5.0/install-sh
fort-1.5.0/README.md
```

and install:

```
cd fort-1.5.0
./configure
make
sudo make install
```

```
apnic@fort:~$ cd fort-1.5.0
apnic@fort:~/fort-1.5.0$ ./configure
make
sudo make installchecking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for gcc... gcc
checking whether the C compiler works... yes
```

```

apnic@fort:~/fort-1.5.0$ make
Making all in src
make[1]: Entering directory '/home/apnic/fort-1.5.0/src'
make all-am
make[2]: Entering directory '/home/apnic/fort-1.5.0/src'
gcc -DHAVE_CONFIG_H -I. -Wall -Wno-cpp -std=gnu11 -O2 -g -I/usr/include/libx
ml2 -g -O2 -MT fort-main.o -MD -MP -MF .deps/fort-main.Tpo -c -o fort-main.o `te
st -f 'main.c' || echo './`main.c
mv -f .deps/fort-main.Tpo .deps/fort-main.Po
gcc -DHAVE_CONFIG_H -I. -Wall -Wno-cpp -std=gnu11 -O2 -g -I/usr/include/libx
ml2 -g -O2 -MT fort-address.o -MD -MP -MF .deps/fort-address.Tpo -c -o fort-addr
ess.o `test -f 'address.c' || echo './`address.c
mv -f .deps/fort-address.Tpo .deps/fort-address.Po
gcc -DHAVE_CONFIG_H -I. -Wall -Wno-cpp -std=gnu11 -O2 -g -I/usr/include/libx
ml2 -g -O2 -MT fort-algorithm.o -MD -MP -MF .deps/fort-algorithm.Tpo -c -o fort-

```

```

make[1]: Nothing to be done for 'all-am'.
make[1]: Leaving directory '/home/apnic/fort-1.5.0'
apnic@fort:~/fort-1.5.0$ sudo make install
[sudo] password for apnic:
Making install in src
make[1]: Entering directory '/home/apnic/fort-1.5.0/src'
make[2]: Entering directory '/home/apnic/fort-1.5.0/src'
  /usr/bin/mkdir -p '/usr/local/bin'
  /usr/bin/install -c fort '/usr/local/bin'
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/apnic/fort-1.5.0/src'
make[1]: Leaving directory '/home/apnic/fort-1.5.0/src'
Making install in man
make[1]: Entering directory '/home/apnic/fort-1.5.0/man'
make[2]: Entering directory '/home/apnic/fort-1.5.0/man'
make[2]: Nothing to be done for 'install-exec-am'.
  /usr/bin/mkdir -p '/usr/local/share/man/man8'

```

- Note:

- there are other install options listed in the vendor [documentation](#)
- the debian package comes with a systemd service, which allows us to run it as a daemon

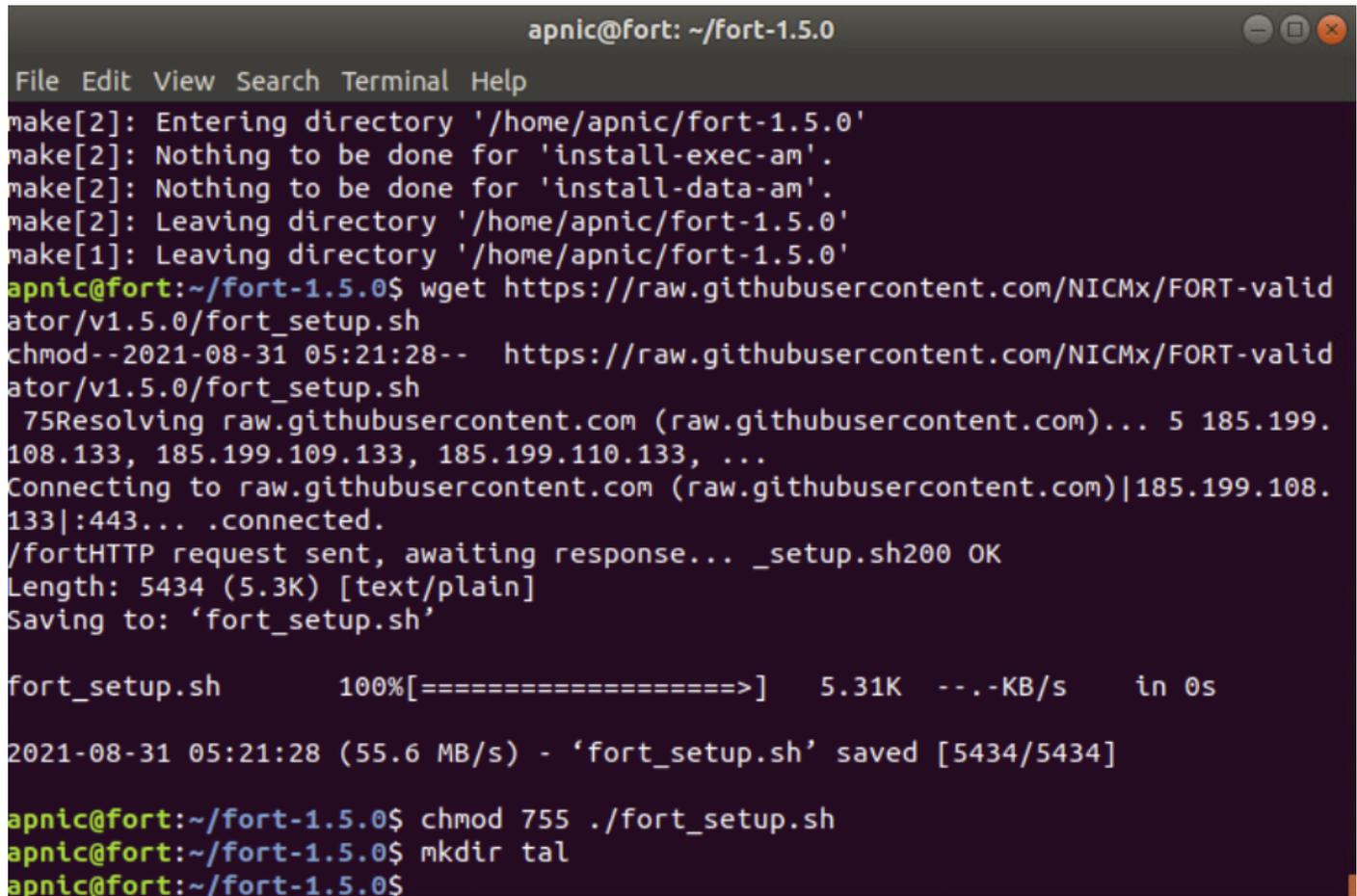
#### 4. Install the 5 Regional Internet Registrars (RIR) trust anchor locators (TALs)

Download the script to set this up

```
wget https://raw.githubusercontent.com/NICMx/FORT-validator/v1.5.0/fort_setup.sh
chmod 755 ./fort_setup.sh
```

Create a directory to store the TALs

```
mkdir tal
```



```
apnic@fort: ~/fort-1.5.0
File Edit View Search Terminal Help
make[2]: Entering directory '/home/apnic/fort-1.5.0'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/apnic/fort-1.5.0'
make[1]: Leaving directory '/home/apnic/fort-1.5.0'
apnic@fort:~/fort-1.5.0$ wget https://raw.githubusercontent.com/NICMx/FORT-validator/v1.5.0/fort_setup.sh
chmod--2021-08-31 05:21:28-- https://raw.githubusercontent.com/NICMx/FORT-validator/v1.5.0/fort_setup.sh
 75Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 5 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... .connected.
/forthHTTP request sent, awaiting response... _setup.sh200 OK
Length: 5434 (5.3K) [text/plain]
Saving to: 'fort_setup.sh'

fort_setup.sh      100%[=====>]    5.31K  --.-KB/s   in 0s

2021-08-31 05:21:28 (55.6 MB/s) - 'fort_setup.sh' saved [5434/5434]

apnic@fort:~/fort-1.5.0$ chmod 755 ./fort_setup.sh
apnic@fort:~/fort-1.5.0$ mkdir tal
apnic@fort:~/fort-1.5.0$
```

To install the ARIN TAL you need to read the ARIN Relying Party Agreement. This can be downloaded from <https://www.arin.net/resources/manage/rpki/rpa.pdf>

Run the downloaded script

```
./fort_setup.sh tal/
```

You need to agree/accept ARIN's Relying Party Agreement, type  to continue.

```
apnic@fort:~/fort-1.5.0$ ./fort_setup.sh tal/

Please download and read ARIN Relying Party Agreement (RPA) from https://www.arin.net/resources/manage/rpki/rpa.pdf

Once you've read and if you agree ARIN RPA, type "yes" to proceed with ARIN's TAL download: yes
```

If you get an **ERROR 404: Not Found** for downloading RIPE tal. this is because some times the script has the wrong name for the RIPE TAL and needs to be updated.

```
Fetching RIPE TAL...
--2021-08-31 04:21:44-- https://raw.githubusercontent.com/NICMx/FORT-validator/master/examples/tal/ripe.tal
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.110.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 404 Not Found
2021-08-31 04:21:44 ERROR 404: Not Found.

Couldn't fetch RIPE TAL, try again.
apnic@fort:~/fort-1.5.0$
```

Type the following to update the name and re-run the script:

```
sed -i 's/ripe/ripe\n-ncc/g' fort_setup.sh
./fort_setup.sh tal/
```

```
Aug 31 04:27:00 ERR [Validation]: Error requesting URL https://raw.githubusercontent.com/NICMx/FORT-validator/master/examples/tal/ripe.tal (received HTTP code 404): The requested URL returned error: 404
Couldn't fetch 'tal/ripe.tal'.
apnic@fort:~/fort-1.5.0$ sed -i 's/ripe/ripe\n-ncc/g' fort_setup.sh
apnic@fort:~/fort-1.5.0$ ./fort_setup.sh tal/

Please download and read ARIN Relying Party Agreement (RPA) from https://www.arin.net/resources/manage/rpki/rpa.pdf

Once you've read and if you agree ARIN RPA, type "yes" to proceed with ARIN's TAL download: yes
```

If the script is successful, it should download the five RIR TALs, the local repo directory is created, and **fort-config.json** file is created with directory path to the TALs and the local repo.

```
apnic@fort: ~/fort-1.5.0
File Edit View Search Terminal Help
mkdir: created directory '/tmp/fort'
mkdir: created directory '/tmp/fort/repository'

-----
----- Success! -----
-----

- The five RIRs TAL's were downloaded to '/home/apnic/fort-1.5.0/tal'.
- The directory /tmp/fort/repository was created, so it can be used as the local
  repository.
- The configuration file '/home/apnic/fort-1.5.0/fort-config.json' was created w
  ith the following content:
{
  "local-repository": "/tmp/fort/repository",
  "tal": "/home/apnic/fort-1.5.0/tal"
}

- This configuration file can be utilized with FORT validator, e.g.:
$ fort -f "/home/apnic/fort-1.5.0/fort-config.json"
- Or its members can be utilized as FORT validator arguments, e.g.:
$ fort --tal "/home/apnic/fort-1.5.0/tal" --local-repository "/tmp/fort/reposit
ory"

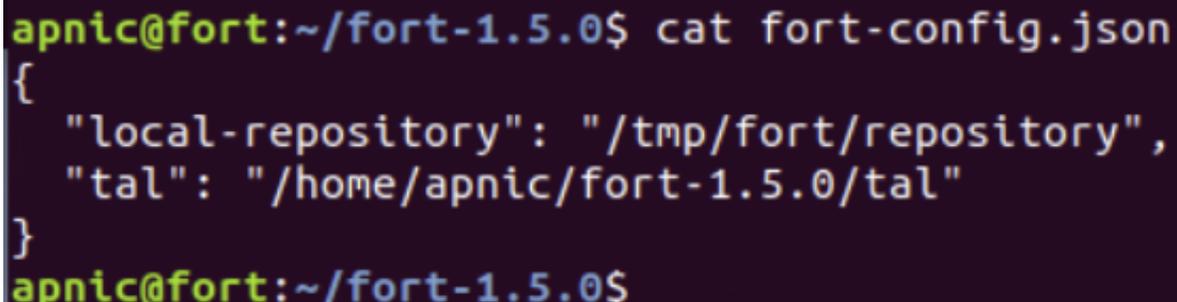
apnic@fort:~/fort-1.5.0$
```

- Other options that can be added to the **fort-config.json** file are:
  - **validation**: time (in seconds) the Validator should wait after updating and validating the ROA cache before updating again from the global repo.
  - **refresh**: time (in seconds) the RTR client (router) has to wait before trying to poll the Validator cache ([RFC8210](#) default 3600 seconds).
  - **retry**: time (in seconds) the RTR client should wait before retrying after a failed refresh of the cache (RFC8210 default 600 seconds).
  - **expire**: time (in seconds) the RTR client can use its validated ROA cache if cannot refresh the data, after which it should discard (RFC8210 default 7200 seconds).
  - **output**: print validated ROAs to a CSV file

```
{
  "tal": "/etc/fort/tal",
  "server": {
    "address": "192.168.30.X",
    "port": "8323",
    "interval": {
      "validation": 900,
      "refresh": 900,
      "retry": 600,
      "expire": 7200
    }
  },
  "output": {
    "roa": "/tmp/fort/fort.csv"
  }
}
```

5. View the FORT configuration file.

```
cat fort-config.json
```



```
apnic@fort:~/fort-1.5.0$ cat fort-config.json
{
  "local-repository": "/tmp/fort/repository",
  "tal": "/home/apnic/fort-1.5.0/tal"
}
apnic@fort:~/fort-1.5.0$
```

6. View the FORT TAL files and rename the rip-ncc.tal to ripe.tal.

```
ls -lash tal/
mv tal/ripe-ncc.tal tal/ripe.tal
```

```

apnic@fort:~/fort-1.5.0$ ls -lash tal/
total 28K
4.0K drwxrwxr-x 2 apnic apnic 4.0K Sep  1 02:43 .
4.0K drwxr-xr-x 7 apnic apnic 4.0K Sep  1 02:43 ..
4.0K -rw-rw-r-- 1 apnic apnic  496 Sep  1 02:43 afrinic.tal
4.0K -rw-rw-r-- 1 apnic apnic  466 Sep  1 02:43 apnic.tal
4.0K -rw-rw-r-- 1 apnic apnic  487 Aug 31 19:47 arin.tal
4.0K -rw-rw-r-- 1 apnic apnic  502 Sep  1 02:43 lacnic.tal
4.0K -rw-rw-r-- 1 apnic apnic  482 Sep  1 02:43 ripe-ncc.tal
  0 -rw-rw-r-- 1 apnic apnic    0 Sep  1 02:43 ripe.tal
apnic@fort:~/fort-1.5.0$ mv tal/ripe-ncc.tal tal/ripe.tal
apnic@fort:~/fort-1.5.0$

```

7. Start the validator (RTR server):

```

fort -f ./fort-config.json --server.address="192.168.30.1#8323" --output.roa="/tmp/fort/roas.csv" &

```

**NOTE:** If a message about **ERR: Waiting thread to start**, rerun the above command

The RPKI to Router Protocol (RTR) server **192.168.30.1** is listening on port **8323** and is printing the validated ROA payload (VRP) to an output file **/tmp/fort/roas.csv**

Confirm FORT is running

```

ps aux | grep fort

```

Review the contents of the **roas.csv** file.

```

more /tmp/fort/roas.csv

```

NOTE: Press **q** to quit from the roas.csv output. If there is no output yet, it means Fort is still working through the initial process of fetching ROAs from the repos and validating, which generally takes a while.

```

apnic@fort:~/fort-1.5.0$ fort -f ./fort-config.json --server.address="192.168.30
.1#8323" --output.roa="/tmp/fort/roas.csv" &
[1] 5451
apnic@fort:~/fort-1.5.0$ Sep  1 04:27:02 WRN: First validation cycle has begun,
wait until the next notification to connect your router(s)

apnic@fort:~/fort-1.5.0$ ps aux | grep fort
apnic      5451  158  0.7 750084 60800 pts/4    Sl   04:27   1:03 fort -f ./fort-
config.json --server.address=192.168.30.1#8323 --output.roa=/tmp/fort/roas.csv
apnic      5537  0.0  0.0   8900   724 pts/4    S+  04:27   0:00 grep --color=au
to fort
apnic@fort:~/fort-1.5.0$ more /tmp/fort/roas.csv
apnic@fort:~/fort-1.5.0$ more /tmp/fort/roas.csv
apnic@fort:~/fort-1.5.0$ Sep  1 04:31:15 WRN: First validation cycle successfull
y ended, now you can connect your router(s)

```

This will display the validated Route Origin Authorisation (ROA) payload

```

apnic@fort:~/fort-1.5.0$ more /tmp/fort/roas.csv
apnic@fort:~/fort-1.5.0$ more /tmp/fort/roas.csv
apnic@fort:~/fort-1.5.0$ Sep  1 04:31:15 WRN: First validation cycle successfull
y ended, now you can connect your router(s)

apnic@fort:~/fort-1.5.0$ more /tmp/fort/roas.csv
ASN,Prefix,Max prefix length
AS205897,2c0f:5400::/48,48
AS205897,154.70.200.0/24,24
AS205897,154.70.202.0/24,24
AS205897,154.70.203.0/24,24
AS205897,154.70.204.0/24,24
AS205897,154.70.205.0/24,24
AS205897,154.70.206.0/24,24
AS205897,154.70.207.0/24,24
AS205897,2c0f:5400::/32,32
AS328799,2c0f:5400:1::/48,48
AS328799,154.70.201.0/24,24
AS328578,102.68.141.0/24,25
AS328578,2001:43f8:1440::/48,48
AS328578,2c0f:2e00::/32,48
AS0,196.60.70.0/23,23
AS0,196.10.141.0/24,24
AS0,196.60.8.0/23,23

```

Check the local cache (of the repository)

```
ls /tmp/fort/repository
```

```
apnic@fort:~/fort-1.5.0$ ls /tmp/fort/repository
0739DB4C  chloe.sobornost.net  repo1.rpki.qs.nu      rpki-repository.nic.ad.jp
2819D8FB  EB240A3F             rpki.afrinic.net     rpki.ripe.net
cb.rg.net  EB354068             rpki.apnic.net        rrdp.arin.net
cc.rg.net  FF2D17F5             rpki.caramelfox.net  rrdp.lacnic.net
apnic@fort:~/fort-1.5.0$
```

8. **[Optional]** If you have two separate Validators installed (for redundancy/code diversity), compare the validated ROA payload outputs for consistency:

- But before you compare the VRPs, you need to sort the output. Example below using `sort` to sort alphanumerically:

```
sort /tmp/fort/roas.csv > fort_sorted.csv
```

To sort the file, but ignoring the header row

```
cd ~/
cat /tmp/fort/roas.csv | { head -n 1 ; sort; } > fort_sorted.csv
```

- Now you can compare the validated ROA outputs, for example Routinator and Fort:

```
diff -u rout_sorted.csv fort_sorted.csv
```

- Discuss any differences with your group mates and instructor.

***Now your validator is ready to feed the validated cache to BGP speaking routers through the RTR (RPKI-to-Router) protocol.***

## Part-2: RTR session

---

### Validator side

Fort can act as an RTR server, to allow RPKI enabled routers to connect to it and fetch the validated cache (ROA cache).

- Based on the configuration file or the FORT start command, the RTR server is listening on `192.168.30.X` (where X is your VM number) and port `8323`
  - The timers can be tweaked to suit your need ([RFC8210](#) has recommendations).
-