Introduction to Artemis Detector

Chris Papastamos

The Artemis Detector Script

- The artemis_detector is a script written in GoLang that mimics the way Artemis detects BGP hijacks.
- The script has two running modes: detect_all and detect_as. We will focus on the second mode since this is the one you are requested to use.
- The detect_all mode will parse the file for hijacks concerning any AS
- The detect_as mode will parse the file for hijacks concerning the AS specified using the –asn flag.

The Artemis Detector Script

- **The script requires the following flags to run**
- **–updates ⟨File/Dir⟩: *The BGP updates file/directory***
- **–input type ⟨"file"/"directory"⟩: *The type of input you are providing the tool with (default:"file")***
- **–prefixes ⟨prefixes file.txt⟩: *The file containing the legal prefixes of each AS.***
- **–output ⟨output file.csv⟩: *The file where the detector will write the final detected hijacks (will be created if not exists)***

Detecting a Hijack

You can detect a hijack with one of the three following ways:

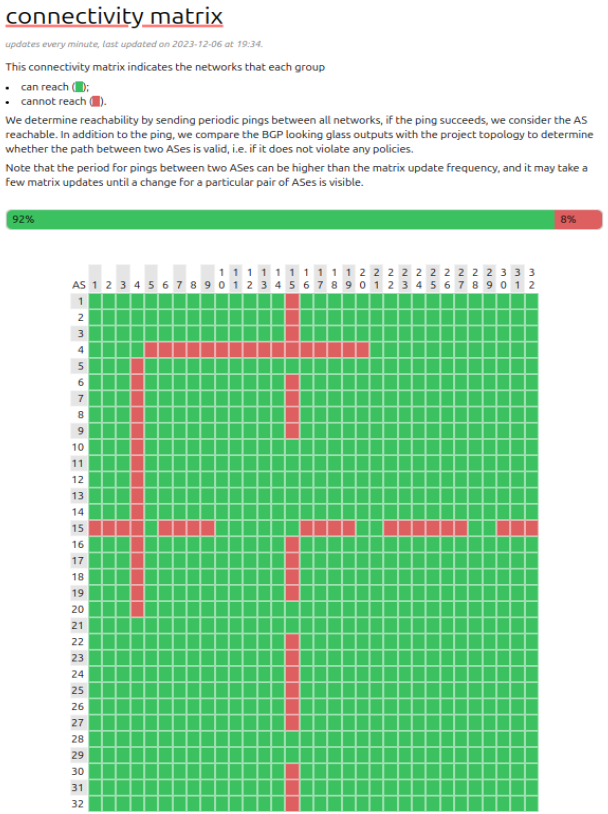
- **Using the Connectivity Matrix**
- **Executing Measurements from neighbor ASes**
- **Looking for your prefix through the Looking Glass**

Detecting a Hijack using the Connectivity Matrix

Using the connectivity matrix to detect hijacks is pretty straight forward. When a hijack occurs, many ASes will lose connectivity to the hijacked AS. That is the reason why these lines appear at the Connectivity Matrix. By tracking the intersection of these tangent lines, you can detect the hijacked AS (The tangent lines should concern the same AS).

In this scenario, the AS 4 and AS 15 are hijacked.

Your AS has been hijacked, you should now mitigate the attack!



Collecting information about the hijack

- **After recognizing that your AS has been hijacked, you should run artemis_detector and use all the required monitor flags in order to collect information about the hijack.**
- **Copy the directory containing the monitor output of each AS and the as_prefixes file. Make sure to move the as_prefixes.txt file out of the folder containing the monitor files.**
- **You are now ready to run artemis on your monitor files!**

Collecting information about the hijack

After recognizing that your AS has been hijacked, you should run Artemis in detect_as mode and the monitor files directory as input. The script will process each file and show statistics about the analysis

```
./artemis_detector detect_as \
--updates test2 \
--input_type directory \
--prefixes as_prefixes.csv \
--output mini-internet_hijacks.csv \
--asn 4 \

Generating Peer Graph...
Generating Patricia Tree...
Processing file: 10_output.csv
  0% |                                                          | (0/0, 0 it/hr) [0s:0s]
Processing file: 11_output.csv
  0% |                                                          | (0/0, 0 it/hr) [0s:0s]
Processing file: 12_output.csv
```

```
  0% |                                                          | (0/0, 0 it/hr) [0s:0s]
Processing file: 8_output.csv
  0% |                                                          | (0/0, 0 it/hr) [0s:0s]
Processing file: 9_output.csv
  0% |                                                          | (0/0, 0 it/hr) [0s:0s]
We have detected 15 hijacks:
        - 0 ongoing hijacks
        - 15 terminated hijacks
        - 0 mitigated prefixes
```

Collecting information about the hijack

The script will output the information of each hijack in the csv file specified as the output file. In this file you can find information like the time of the hijack, the hikacker asn, the hijack type, the hijack type and the hijack state.

<u>Note</u>: even though one hijack event occurred, BGP will advertise the prefix multiple times making multiple entries appear in the output.

artemis_go > mini-internet_hijacks.csv

```
prefix,origin as,hijack type,hijacker asn,time started,time of last update,time ended,state,duration
4.0.0.0/8,5,E|0|-|-,5,1701882367.608717,1701882368.000000,1701882368.000000,Withdrawn,0.016667
4.0.0.0/8,5,E|0|-|-,5,1701882367.596782,1701885952.000000,1701885952.000000,Withdrawn,59.750000
4.0.0.0/8,5,E|0|-|-,5,1701882367.624818,1701882368.000000,1701882368.000000,Withdrawn,0.016667
4.0.0.0/8,5,E|0|-|-,5,1701882367.131501,1701882368.000000,1701882368.000000,Withdrawn,0.016667
4.0.0.0/8,5,E|0|-|-,5,1701882367.379741,1701882368.000000,1701882368.000000,Withdrawn,0.016667
4.0.0.0/8,5,E|0|-|-,5,1701882367.349355,1701885952.000000,1701885952.000000,Withdrawn,59.750000
4.0.0.0/8,5,E|0|-|-,5,1701882367.557364,1701882368.000000,1701882368.000000,Withdrawn,0.016667
4.0.0.0/8,5,E|0|-|-,5,1701882367.234699,1701882368.000000,1701882368.000000,Withdrawn,0.016667
4.0.0.0/8,5,E|0|-|-,5,1701882367.254910,1701882368.000000,1701882368.000000,Withdrawn,0.016667
4.0.0.0/8,5,E|0|-|-,5,1701882367.725775,1701885952.000000,1701885952.000000,Withdrawn,59.750000
4.0.0.0/8,5,E|0|-|-,5,1701882367.481537,1701882368.000000,1701882368.000000,Withdrawn,0.016667
4.0.0.0/8,5,E|0|-|-,5,1701882367.371470,1701882368.000000,1701882368.000000,Withdrawn,0.016667
4.0.0.0/8,5,E|0|-|-,5,1701882367.229423,1701882368.000000,1701882368.000000,Withdrawn,0.016667
4.0.0.0/8,5,E|0|-|-,5,1701882367.562086,1701885952.000000,1701885952.000000,Withdrawn,59.750000
4.0.0.0/8,5,E|0|-|-,5,1701882367.119766,1701882368.000000,1701882368.000000,Withdrawn,0.016667
```

Mitigating the hijack

- **There are multiple ways to mitigate (and prevent) a hijack**
- **In order to mitigate the hijack, you will advertise the two sub-prefixes that your original prefix contains.**
- **Use basic subnetting techniques to calculate the two sub-prefixes.**

Mitigating the hijack

- After calculating the sub-prefixes, you will need to advertise each one at every edge router to restore connectivity
- Verify your changes by looking at the output of the `show run` command
- After successfully re-advertising the sub-prefixes, execute a measurement from a previously affected AS and verify that connectivity is restored

```
Hello, this is FRRouting (version 8.2.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

EAST_router# conf t
EAST_router(config)# ip route 2.0.0.0/9 Null0
EAST_router(config)# ip route 2.128.0.0/9 Null0
EAST_router(config)# router bgp 2
EAST_router(config-router)# network 2.0.0.0/9
EAST_router(config-router)# network 2.128.0.0/9
EAST_router(config-router)# exit
EAST_router(config)# exit
EAST_router#
```