Run

Howard Network Topology Project

1. traceroute.py

This is the main script for IP scanning and route tracing.

```
parser.add_argument('-m', '--mode', type = str, default = None, help = 'public or internal')
parser.add_argument('-s', '--start', type = str, default = '0.0.0.0', help = 'start ip to scan')
parser.add_argument('-e', '--end', type = str, default = '255.255.255.255', help = 'end ip to scan')
parser.add_argument('-t', '--thread', type = int, default = 8, help = 'number of threads')
parser.add_argument('-r', '--resume', action = 'store_true', default = False, help = 'resume from backup file')
parser.add_argument('-f', '--flag', action = 'store_true', default = False, help = 'reset flags')
parser.add_argument('-v', '--verbose', action = 'store_true', default = False, help = 'true if verbose without prog bar')
parser.add_argument('-z', '--z', type = str, default = '0,1', help = 'last segment setup')
```

There are some control arguments to run it.

-m (or --mode): `public` for Public IP scan, `internal` for Internal IP scan

-s (or --start): limit scanning IP range with start.

-e (or --end): limit scanning IP range with end.

-t (or --thread): number of threads

-r (or --resume): if set, scan resumes from the interrupted point.

-f (or --flag): it ignores formerly flagged IPs.

-v (or --verbose): if set, detailed messages are output instead of progress bar.

-z (or --z): z condition. default value is `0,1`. it is a pair of two integers delimited by comma: `a,b`. it means that z starts from a and increases by step b. for example, if this parameter is set as `1,2`, then all odd numbers of z will be scanned. if it is set as `0,5`, then all multiples of five will be scanned.

Example

python traceroute.py -m internal -r -s 10.10.3.0 -e 10.10.255.255 -z 1,255

This is a resumed scan. (see parameter -r)

It scans internal IPs. (see parameter -m)

It scans from 10.10.3.10 to 10.10.255.255. (see parameters -s and -e)

Only the IPs with the last segment z=1 are scanned. (see parameter -z)

2. visualize.py

This is the visualization script.

It has only one argument -m (or --mode). It denotes the visualization target.

Examples

python visualize.py -m whole

It outputs vis_whole.jpg which is a whole topology diagram.

python visualize.py -m public

It outputs vis_public.jpg which is a topology of Public Howard IPs.

python visualize.py -m bridge

It outputs vis_bridge.jpg which shows the connections between internal and public IPs.

python visualize.py -m gate

It outputs vis_gate.jpg which shows the important IPs with many connections.

python visualize.py -m calc

It prints the distribution of IP node linkages from our discoveries.