RIPE 85
Belgrade, Serbia
24 - 28 October 2022

Kathará
A Lightweight and Scalable Network Emulation System

Mariano Scazzariello

mariano.scazzariello@uniroma3.it

What is it?

Kathará is a network emulation system

Allows users to run real network software in a virtual environment in order to perform tests and experiments on networks

Used worldwide by many universities and companies

- Including some big Internet players

Website: kathara.org

An Open-Source Project

Kathará is completely open-source, and written in Python

The main repository is hosted on GitHub:
github.com/KatharaFramework/Kathara

The project numbers:

- Used by more than **20** courses in more than **10** universities worldwide
- **>50k downloads**
- Almost 1k commits
- 16 contributors around the world
- 220 stars

Support and Compatibility

Kathará is available on all the main Operating Systems

- As far as we know, the only network emulator supporting all of them

We also ship a Python package available on PyPI:
pypi.org/project/kathara/

Complete installation guides can be found in the Wiki:
github.com/KatharaFramework/Kathara/wiki/Installation-Guides

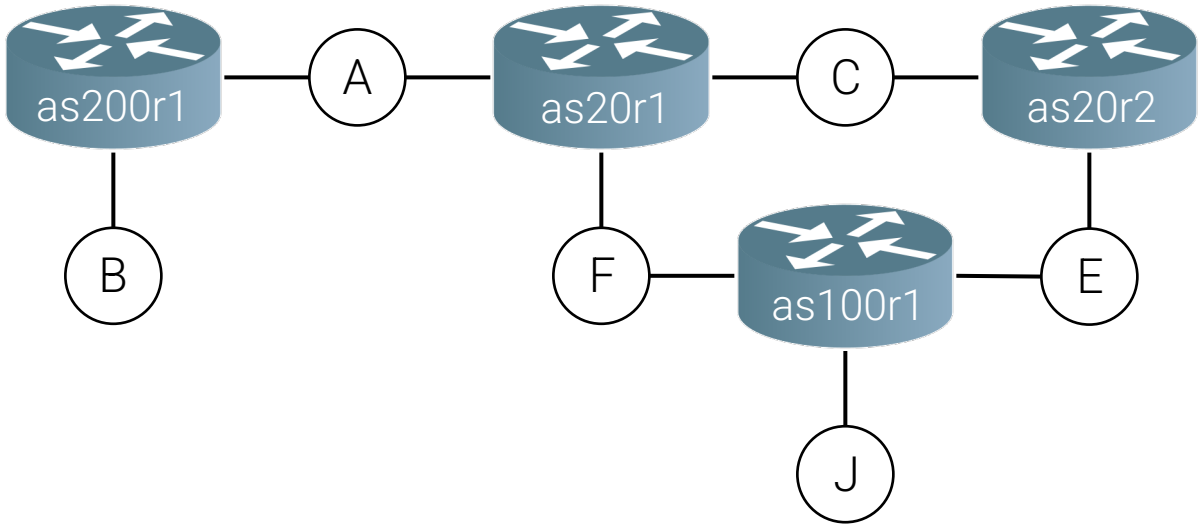
Windows Linux macOS python Package Index

Network Scenario

Simple configuration language to describe a network scenario

A network scenario is represented as a directory, containing:

- A file with the network topology (**lab.conf**)
- For each device, files and folders containing the real configuration of that device



Lab directory structure

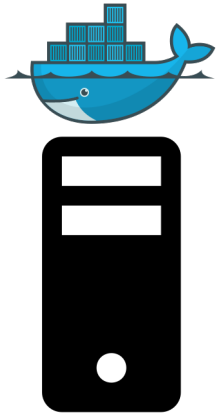
```
|-- as100r1
|   \-- etc
|       \-- quagga
|           |-- bgpd.conf
|           \-- daemons
|-- as100r1.startup
|-- as200r1
|   \-- etc
|       \-- quagga
|           |-- bgpd.conf
|           \-- daemons
|-- as200r1.startup
|-- as20r1
|   \-- etc
|       \-- quagga
|           |-- bgpd.conf
|           \-- daemons
|-- as20r1.startup
|-- as20r2
|   \-- etc
|       \-- quagga
|           |-- bgpd.conf
|           \-- daemons
|-- as20r2.startup
\-- lab.conf
```

lab.conf content

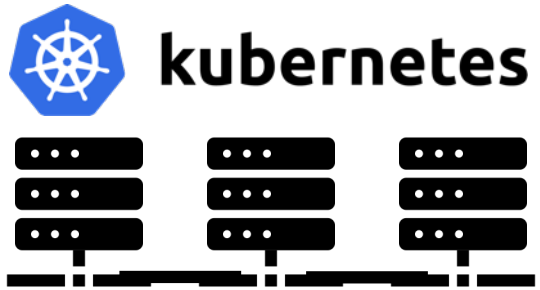
```
as20r1[0]="A"
as20r1[1]="F"
as20r1[2]="C"

as20r2[0]="E"
as20r2[1]="C"

as200r1[0]="A"
as200r1[1]="B"

as100r1[0]="E"
as100r1[1]="F"
as100r1[2]="J"
```

Scalability

More than 1000 devices on a common laptop (6C-12T/16GB RAM)



More than 2000 devices on a single server (32C-64T/512GB RAM)



No limits on a Kubernetes cluster

Use Cases

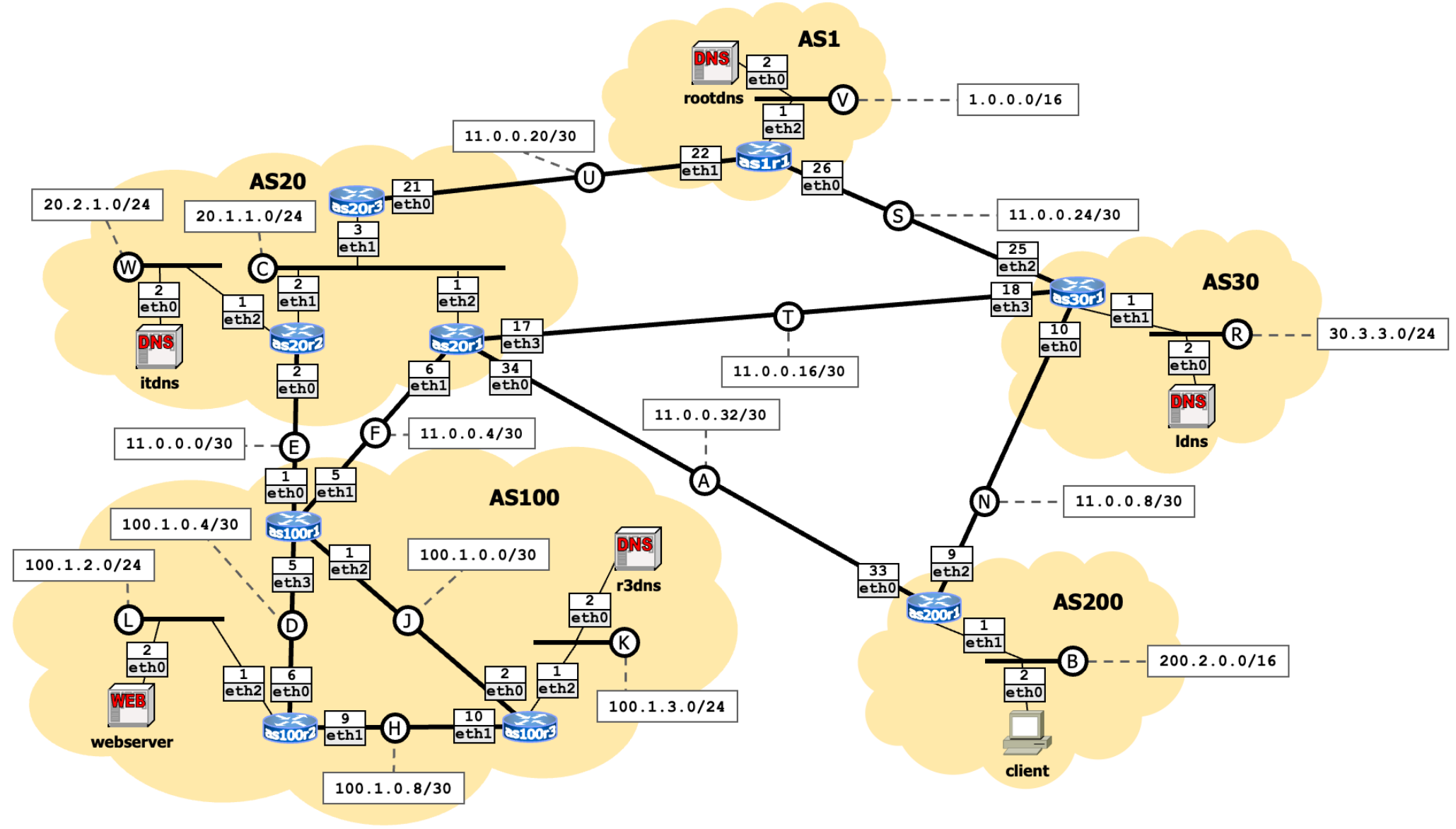
Configuration Testing

Deploy arbitrary complex scenarios for testing configurations before deploying them in the production network

Interoperability testing between configurations, versions and different implementations

Leveraging on L2 virtual networks, it is also possible to run non-IP-based protocols (*e.g.*, IS-IS)

Configuration Testing – Example Scenario



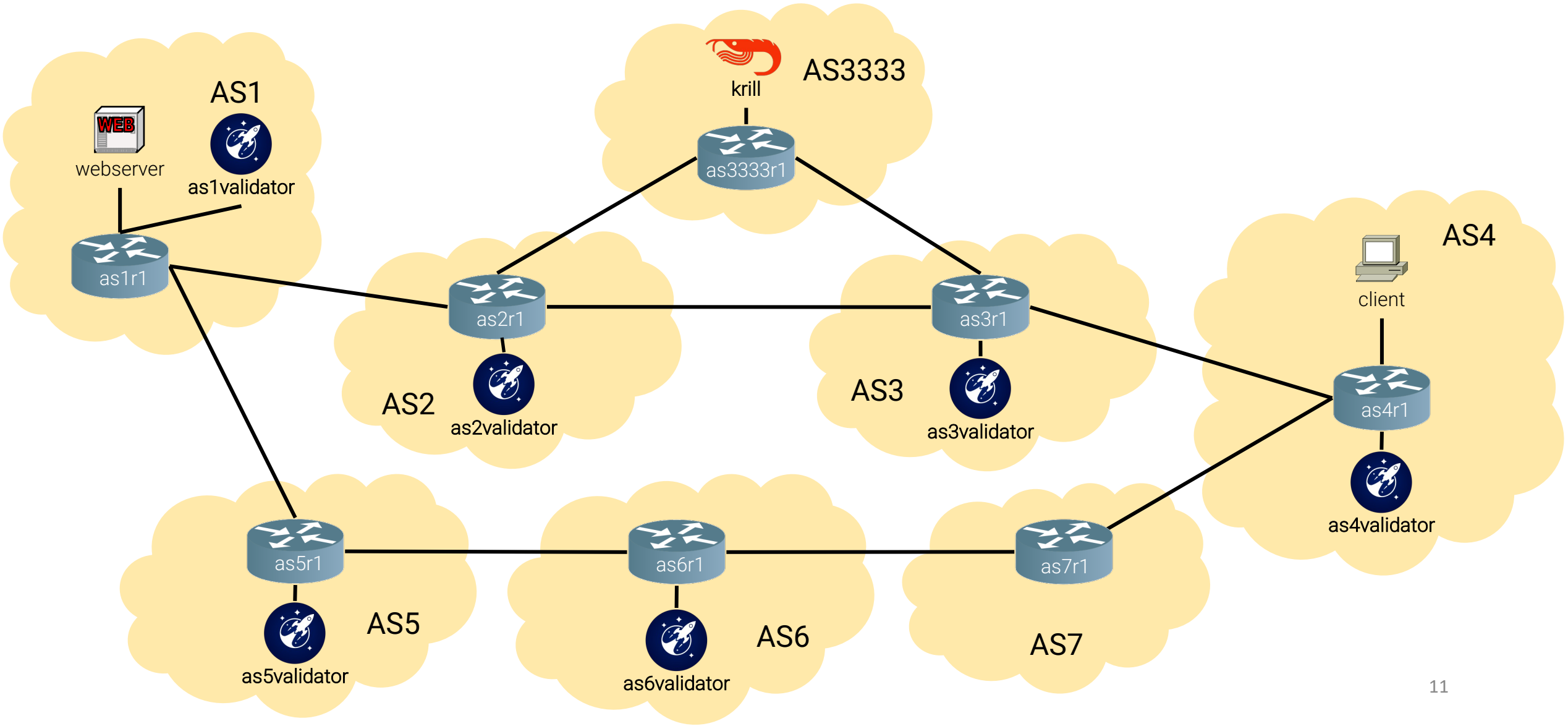
What-If Scenarios for Security Assessment

Possibility to test what-if scenarios for security assessment

Analyze how configuration changes are reflected inside and outside your network

Analyze possible attacks and test the efficiency of the implemented countermeasures

What-If Scenarios for Security Assessment



root@webserver: /

```
20:07:51.864945 IP 193.201.0.2 > 193.204.0.2: ICMP echo rep
ly, id 1, seq 509, length 64
20:07:52.888911 IP 193.204.0.2 > 193.201.0.2: ICMP echo req
uest, id 1, seq 510, length 64
20:07:52.888939 IP 193.201.0.2 > 193.204.0.2: ICMP echo rep
ly, id 1, seq 510, length 64
20:07:53.912813 IP 193.204.0.2 > 193.201.0.2: ICMP echo req
uest, id 1, seq 511, length 64
20:07:53.912859 IP 193.201.0.2 > 193.204.0.2: ICMP echo rep
ly, id 1, seq 511, length 64
20:07:54.936847 IP 193.204.0.2 > 193.201.0.2: ICMP echo req
uest, id 1, seq 512, length 64
20:07:54.936880 IP 193.201.0.2 > 193.204.0.2: ICMP echo rep
ly, id 1, seq 512, length 64
20:07:55.964926 IP 193.204.0.2 > 193.201.0.2: ICMP echo req
uest, id 1, seq 513, length 64
20:07:55.964952 IP 193.201.0.2 > 193.204.0.2: ICMP echo rep
ly, id 1, seq 513, length 64
```

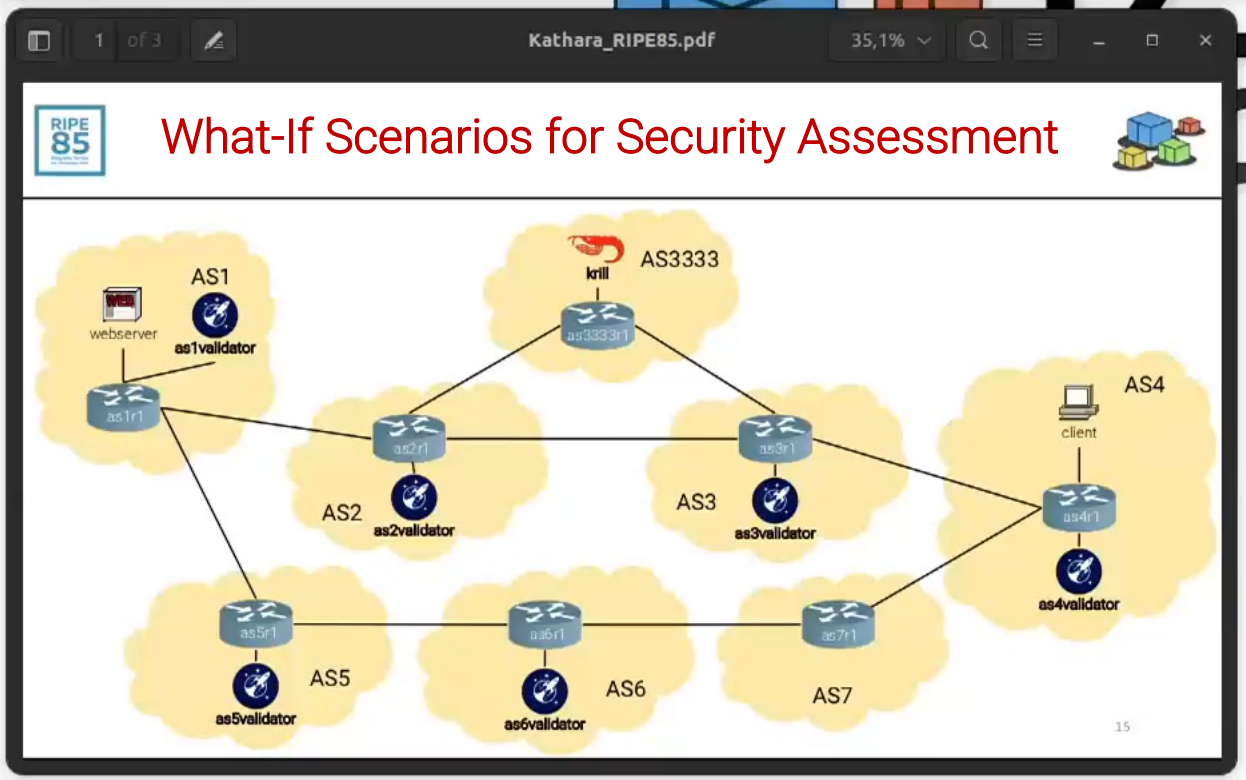
root@as3r1: /

```
20:07:53.105266 IP 10.3.4.1.bgp > 10.3.4.2.60566: Flags
[.], ack 171, win 506, options [nop,nop,TS val 424364780
2 ecr 3991433174], length 0
20:07:53.105273 IP 10.3.4.2.60566 > 10.3.4.1.bgp: Flags
[.], ack 172, win 501, options [nop,nop,TS val 399143317
4 ecr 4243647802], length 0
20:07:53.912758 IP 193.204.0.2 > 193.201.0.2: ICMP echo
request, id 1, seq 511, length 64
20:07:53.912885 IP 193.201.0.2 > 193.204.0.2: ICMP echo
reply, id 1, seq 511, length 64
20:07:54.936781 IP 193.204.0.2 > 193.201.0.2: ICMP echo
request, id 1, seq 512, length 64
20:07:54.936908 IP 193.201.0.2 > 193.204.0.2: ICMP echo
reply, id 1, seq 512, length 64
20:07:55.964866 IP 193.204.0.2 > 193.201.0.2: ICMP echo
request, id 1, seq 513, length 64
20:07:55.964988 IP 193.201.0.2 > 193.204.0.2: ICMP echo
reply, id 1, seq 513, length 64
```

root@as4r1: /

```
Every 2.0s: vtysh -c 'show ip bgp'        as4r1: Thu Oct 20 20:07:55 2022

BGP table version is 7, local router ID is 10.3.4.2, vrf id 0
Default local pref 100, local AS 4
Status codes:  s suppressed, d damped, h history, * valid, > best, = mul
tipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 193.201.0.0/16   10.3.4.1                 150      0 3 2 1 i
*                   10.7.4.1                 150      0 7 6 5 1 i
*> 193.202.0.0/16   10.3.4.1                 150      0 3 2 i
*> 193.203.0.0/16   10.3.4.1               0 150      0 3 i
*> 193.204.0.0/16   0.0.0.0                  0         32768 i
*  193.205.0.0/16   10.3.4.1                 150      0 3 2 1 5 i
*>                  10.7.4.1                 150      0 7 6 5 i
*> 193.206.0.0/16   10.7.4.1                 150      0 7 6 i
```

Kathara_RIPE85.pdf



root@as7r1:/#

root@client: /

```
64 bytes from 193.201.0.2: icmp_seq=497 ttl=60 time=0.318 ms
64 bytes from 193.201.0.2: icmp_seq=498 ttl=60 time=0.394 ms
64 bytes from 193.201.0.2: icmp_seq=499 ttl=60 time=0.209 ms
64 bytes from 193.201.0.2: icmp_seq=500 ttl=60 time=0.276 ms
64 bytes from 193.201.0.2: icmp_seq=501 ttl=60 time=0.180 ms
64 bytes from 193.201.0.2: icmp_seq=502 ttl=60 time=0.201 ms
64 bytes from 193.201.0.2: icmp_seq=503 ttl=60 time=0.263 ms
64 bytes from 193.201.0.2: icmp_seq=504 ttl=60 time=0.304 ms
64 bytes from 193.201.0.2: icmp_seq=505 ttl=60 time=0.299 ms
64 bytes from 193.201.0.2: icmp_seq=506 ttl=60 time=0.202 ms
64 bytes from 193.201.0.2: icmp_seq=507 ttl=60 time=0.238 ms
64 bytes from 193.201.0.2: icmp_seq=508 ttl=60 time=0.183 ms
64 bytes from 193.201.0.2: icmp_seq=509 ttl=60 time=0.201 ms
64 bytes from 193.201.0.2: icmp_seq=510 ttl=60 time=0.193 ms
64 bytes from 193.201.0.2: icmp_seq=511 ttl=60 time=0.279 ms
64 bytes from 193.201.0.2: icmp_seq=512 ttl=60 time=0.238 ms
64 bytes from 193.201.0.2: icmp_seq=513 ttl=60 time=0.219 ms
```

Building a Network Testing Pipeline

Python APIs allow to build complex frameworks on top of Kathará

Operators can build complete network testing pipelines to automatically deploy and assess configuration changes

Vendors can build an integration testing pipeline for supporting protocols and NFs development

Assess Routing Protocols Implementations

Define a methodology and implement a software framework (**Sibyl**) for testing routing protocol implementations

Sibyl leverages on Kathará to automatically deploy fat-tree topologies and test routing protocols in hyper-scale datacenters

- Tested on topologies up to 1.3k routers and 33k interfaces

Automatically computes standard and novel metrics for analyzing the protocols behaviour

Sibyl: a Framework for Evaluating the Implementation of Routing Protocols in Fat-Trees. Caiazzi, Scazzariello, Alberro, Ariemma, Castro, Grampin, Di Battista. NOMS 2022. Available here.

Summary

Deploy more than 1K devices on a common laptop

No scalability constraints with the distributed mode

Possibility to emulate networks for operational and research purposes with support to all major routing protocols and suites

Allow to build frameworks for testing implementations and configurations of generic network scenarios

Contacts

Roma Tre – Computer Networks and Security Research Group

Mariano Scazzariello

mariano.scazzariello@uniroma3.it

Tommaso Caiazzi

tommaso.caiazzi@uniroma3.it

Lorenzo Ariemma

lorenzo.ariemma@uniroma3.it

github.com/KatharaFramework/Kathara

linkedin.com/company/kathara

contact@kathara.org

Website