Universität Münster

GMI-AIMS-5 Workshop

Cloud-telescopes

Ongoing work

Nils Kempen

13 February 2025

University of Münster - NetSec Research Group

living.knowledge

Network telescope (mis)-adventures

Idea(s)

- Different telescopes/ vantage-points provide different views
- Understanding which is best for specific observations
- Cloud-based approaches seem promising
  - Still unclear what the best way to operate them is
  - e.g. Holding time of an IP Address,
  - VM configuration,
  - economic perspective

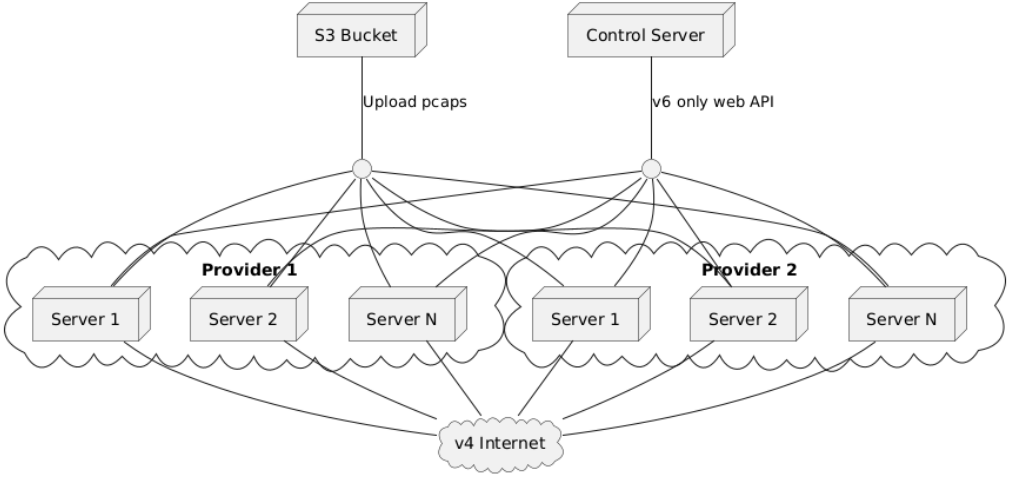
When do we need which lense?

- *current literature™* dosn't provide clear answers yet

Approach - data collection

- Build a distributed, multi-cloud network telescope
  - configurable lifetime
  - provider agnostic
  - variable size
- My original idea → Go program using provider SDK's
- Hackathon idea → Use Terraform scripts to deploy servers

Approach - data collection



Hackathon results

Provider	Cost (IP/M)	Approach
DigitalOcean	4.0$	One VM per IP
OVH	1.8$	Leasing subnet
AWS	7.5$	One VM per IP
Azure	9.0$	One VM per IP
Azure	4.8$	Load balancer
GCP	8.5$	One VM per IP
GCP	5.4$	Load balancer
Alibaba	3.8$	VM with multiple IPs
Vultr	3.5$	One VM per IP

Hackathon results

- Fusion of approaches
- Existing setup of Bernhard for Vultr
- mine for DigitalOcean
- Repurpose Sayed's & Ricky's Terraform code for passive monitoring
- **Working nodes:** Vultr 29 VMs, DigitalOcean 42 VMs,
  AWS 60 VMs, Azure 76 VMs, GCP 109 VMs → 316 VMs/ IPs
- **Cost:** DO 5,6$/D, Vultr 4,8$/D, GCP 42$/D, AWS 54$/D, Azure 210$/D

Hackathon results

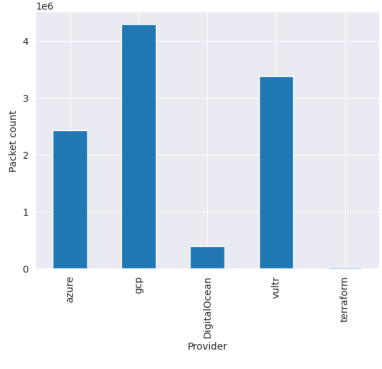


Figure: # of packets per provider

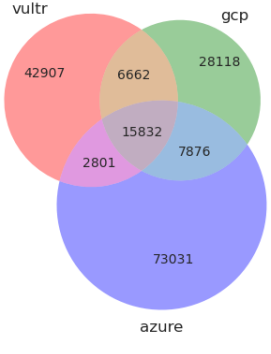


Figure: overlapping source ips

Hackathon results

```
ip
169.254.169.254    441572
103.141.138.254    136554
195.178.110.109     81673
62.210.81.232       81055
195.178.110.41      65317
                    ...
218.110.241.225         1
119.179.35.52           1
92.44.200.110           1
123.245.85.171          1
191.58.49.18            1
Name: count, Length: 104287, dtype: int64
```

Figure: Most common source IPs

What do we see here?

- Local ip
- Scanners
- Hosting-providers
- ?

Hackathon results

```
cc
us    1627002
br    1108868
**     444048
uk     293015
nl     270680
ad     212313
cn     207621
ro     205657
vn     186564
fr     173387
Name: count, dtype: int64
```

Figure: Most common source Countries

```
asn
396982    764248
265928    473611
264332    452453
8075      329906
16509     231781
48090     212298
214295    198548
202425    184212
135905    179102
63949     175747
Name: count, dtype: int64
```

Figure: Most common source ASes

Hackathon results

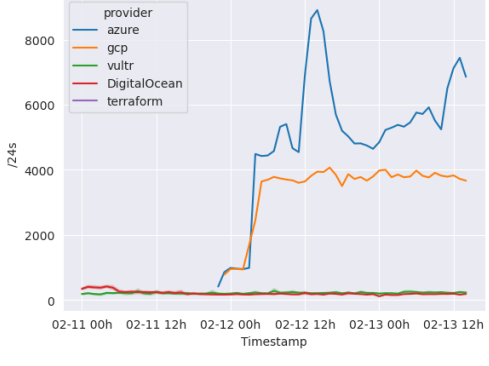


Figure: # of /24s per provider

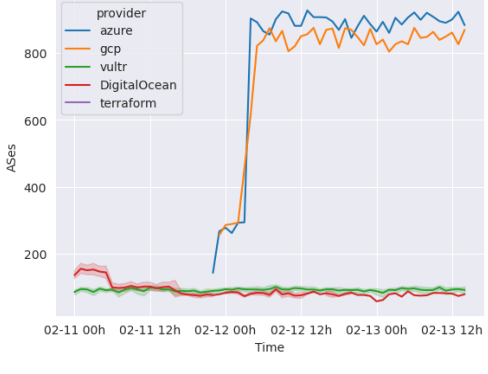


Figure: # of ASes per provider

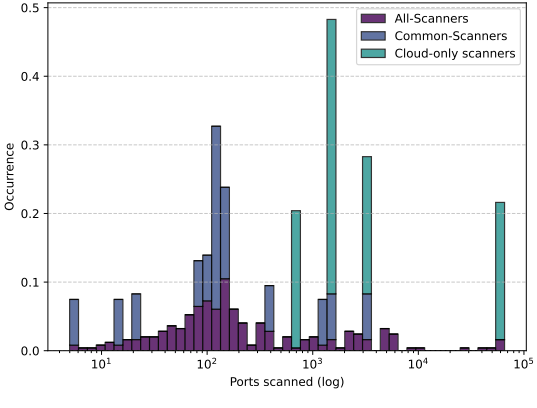
Hackathon learnings

- Deploying a cloud-telescope is hard
  - All cloud-providers work a bit different
  - Destination IPs are often not directly linked to the interface (NAT)
  - Old software
  - Cloud-internal traffic

Future work

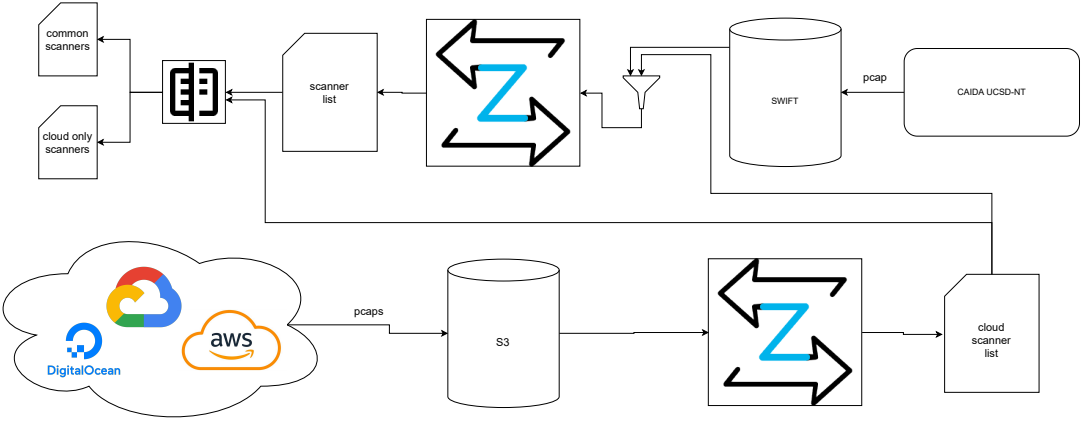
- Fix bugs
- Analyze the data
- Compare with other telescopes/ use them as baseline
  - Identify cloud scanners
  - Look for cloud scanners in other network telescopes → UCSD-NT, …

Future work - inspiration



- If you scan cloud address space you are likely to hit something
- Resource intensive scans could be more focussed and may not be seen in “normal” telescopes.
- Further investigation of cloud-scanner behavior is needed.

Future work - approach



Validation

- What even is a telescope?
  - For cloud approaches we need to investigate what level of interaction we want
  - Save all packets and drop
  - Send RST
  - Complete Handshake
  - Emulate services
- Validate the scanner detection
  - What is a scanner
  - What categories can we build?
  - No clear field-wide definition
- Validate what wee see in the cloud
  - With other telescopes
  - Over time

Questions?