All your Cloud are belong to us

On-and-in the cloud forensics facilitation

Your hosts today

Thomas Chopitea

- Incident responder
- dfTimewolf core developer
- Based in Zurich, Switzerland 🇨🇭
- Twitter: @tomchop_

Theo Giovanna

- libcloudforensics core developer
- Based in Zurich, Switzerland 🇨🇭
- GitHub: @giovannt0

A storm is brewing...

Motivations

- Need to automate Cloud investigations
- Want a ready-to-use investigation environment
  - Manually installing tools takes time...
- Processing evidence in the Cloud is faster

Caveats

- We don’t want to be the Swiss army knife of Cloud
- We want essential functionality with few dependencies

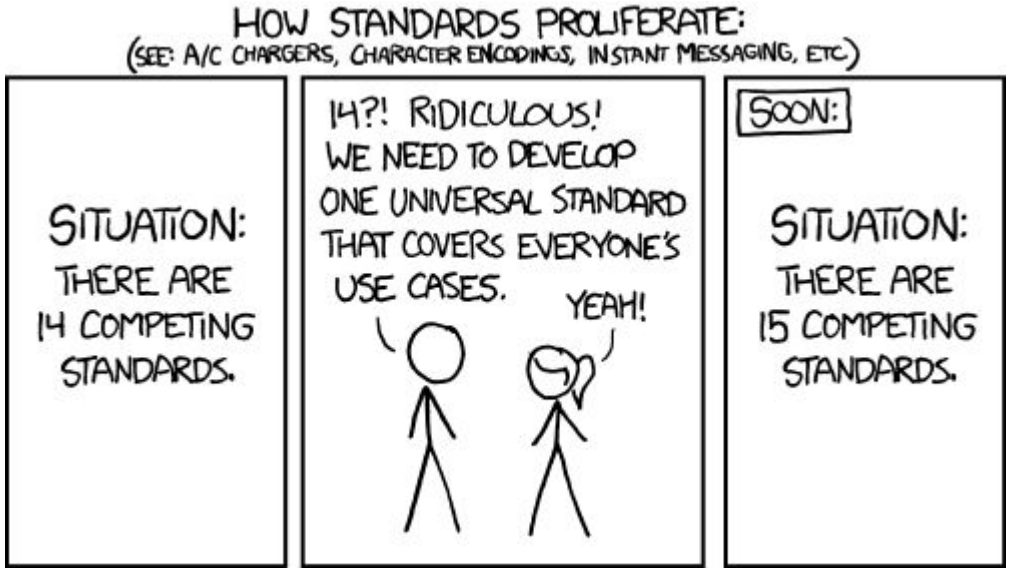
A silver lining - libcloudforensics    

- Python library to interact with different Cloud providers
  - GCP, AWS, Azure, ...
- Lightweight, heavy focus on forensic tasks, e.g.:
  - Copying disks (‘dd in the Cloud’)
  - Spinning up ready-to-use analysis VMs
  - Grabbing all types of logs
- Similar projects
  - Apache libcloud: https://libcloud.apache.org/
  - Forseti Security: https://forsetisecurity.org/

```
2 == "2" // true?
```

- Taxonomy nuances across providers
  - Architecture
  - Naming
  - Capabilities

- Come up with a ***similar*** interface for different providers



https://xkcd.com/927/

Libcloudforensics

Who dis?

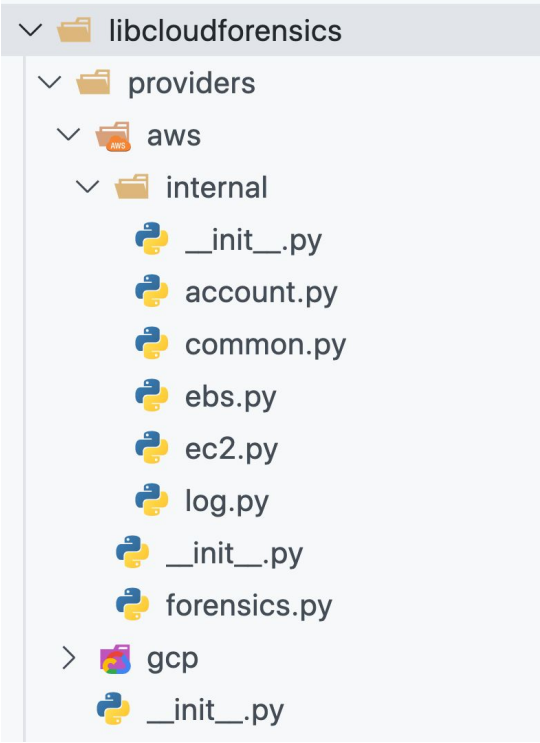
In a nutshell 🌰 🐚

- Python 3 library packaged with a CLI
  - Easy to install and use
  - `$ pip install libcloudforensics && libcloudforensics -h`

- Transparently authenticates and interacts with different cloud providers

- Open source; Apache 2 license

Library architecture



- Primitives for each provider
  - `ListInstances`
  - `ListDisks`
  - `GetVolumeByName`
  - `ListLogs`
  - `LookupEvents`
  - `...`
- Higher-level forensics functionality
  - `CreateVolumeCopy`
  - `CreateDiskCopy`
  - `StartAnalysisVm`

Libcloudforensics

As seen on Google Cloud Platform

Imagine the following scenario...

- A GCE instance hosting a content management system gets owned

- What we want:
  - A forensic copy of the compromised disk(s) belonging to the instance
  - An analysis VM ready to forensicate the disk(s), in a separate project
  - A coffee

- What we don't want:
  - Do any of these tasks manually (except coffee 'cause that's important)

```python
from libcloudforensics.providers.gcp import forensics

# Create a forensic copy of the disk 'disk1'
copy = forensics.CreateDiskCopy(
    src_project='salt-src',
    dst_project='salt-analysis',
    instance_name='instance1',
    zone='us-central-1',
    disk_name='disk1')

# Start an analysis VM 'vm-forensics' for investigation in the project
# 'salt-analysis', and attach the copy created in the previous step.
# Boot disk type and size, numbers of CPU cores are also customizable
analysis_vm, _ = forensics.StartAnalysisVm(
    project='salt-analysis',
    vm_name='vm-forensics',
    zone='us-central-1',
    attach_disk=copy)
```

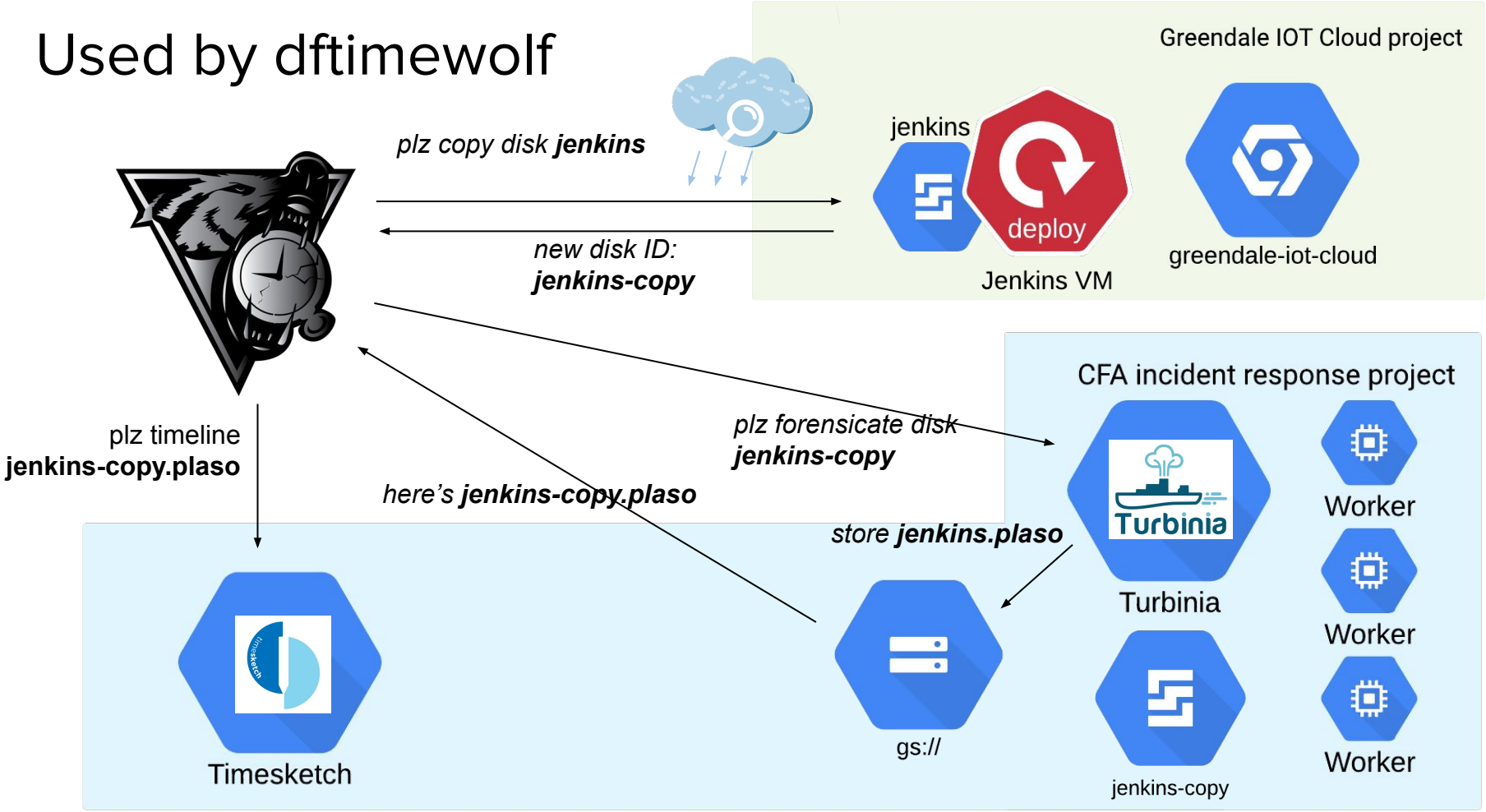
CLI for the win

```
giovannt0@:~/$ libcloudforensics gcp --help
usage: libcloudforensics gcp [-h]
                             project
                             {listinstances,listdisks,copydisk,querylogs,listlogs}
positional arguments:
  project                Source GCP project.
  {listinstances,listdisks,copydisk,querylogs,listlogs}
    listinstances        List GCE instances in GCP project.
    listdisks            List GCE disks in GCP project.
    copydisk             Create a GCP disk copy.
    querylogs            Query GCP logs.
    listlogs             List GCP logs for a project.
optional arguments:
  -h, --help             show this help message and exit
```

We like one-liners

```
~$ libcloudforensics gcp pwned-project copydisk analysis-project pwned-instance us-central-1f
Disk copy completed.
Name: evidence-pwned-instance-202006230-1a01f7c4-copy
~$ libcloudforensics gcp analysis-project startvm analysis-vm us-central-1f --attach_disks=evidence-
pwned-instance-202006230-1a01f7c4-copy
Analysis VM started.
Name: analysis-vm, Started: True
```

Used by dftimewolf



Libcloudforensics

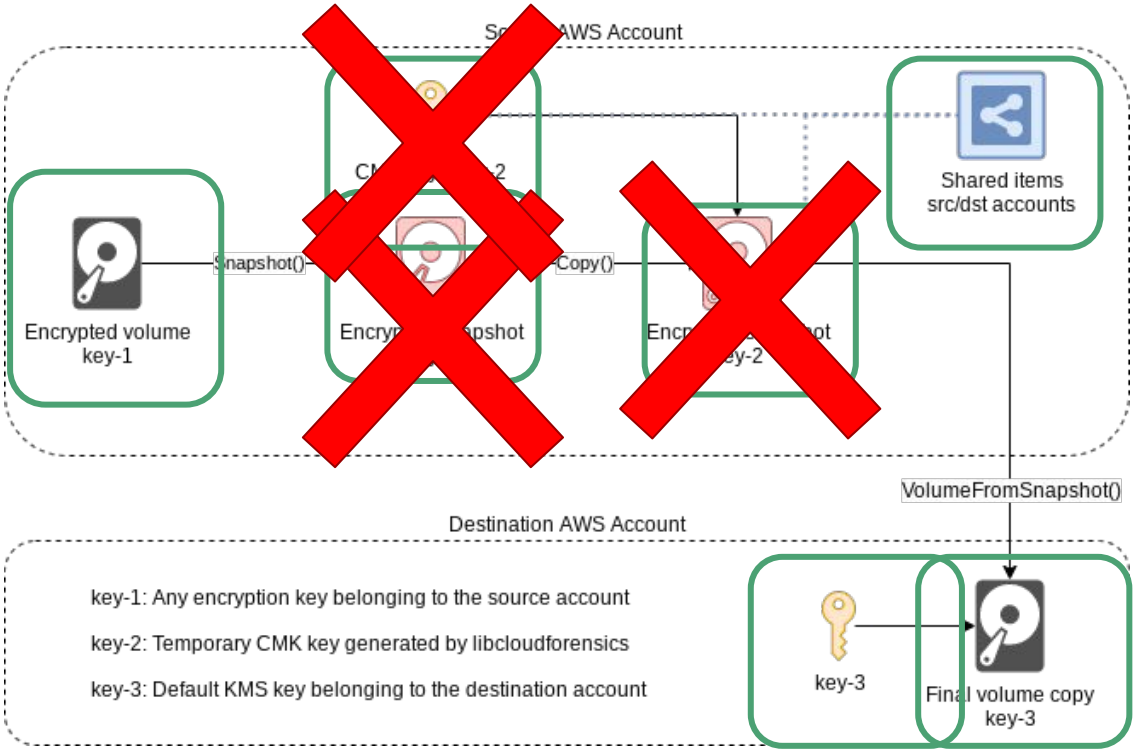
As seen on Amazon Web Services

A similar story

- Let’s take the same scenario, with a few differences:
  - We’re now in AWS EC2
  - The disk(s) to investigate uses AWS EBS Encryption
  - We want to analyze the disk(s) on a different AWS account

- Can we do the previous steps through AWS Web console?
  - Short answer: Yes
  - Long answer: Yes, but you don’t want to

Doing it manually 👎 ☠️



Doing it the cool way 😎

```
~$ libcloudforensics aws us-east-2b copydisk --volume_id=vol-xxx --dst_profile=analysis_account
Starting volume copy ...
Done! Volume vol-yyy successfully created. You will find it in your AWS account under the name evidence-
vol-xxx-snapshot-9bc86af3-copy.
```

Open source

Why open source? 🐧

- We try hard to be platform agnostic
- Never know what we're gonna have to forensicate next
- More diverse feedback (different use-cases, cloud providers)
  - Different use-cases
  - Different experience with cloud providers
- Give something back to the community 😀

👩‍🔬 🧪 Code review experiments 💥

- New library, unfamiliar codebase
- GitHub “Code owners”
- 👀 Two pair of eyes 👀
  - Less reviewer fatigue
  - Less pressure on reviewers
  - More perspectives, more learning opportunities
- Onboard a bigger part of the team
  - Multi-timezone bug-fixes
  - Maintenance responsibilities
  - Code reviews

🚀 CI / CD pipeline 🌙

- We're security engineers, not software engineers!
  - We want to focus on features
  - Automate as much of the QA as possible
- GitHub actions for unit testing
  - Merge with confidence
- Jenkins for e2e testing
  - Detect API changes early, not when you have an incident

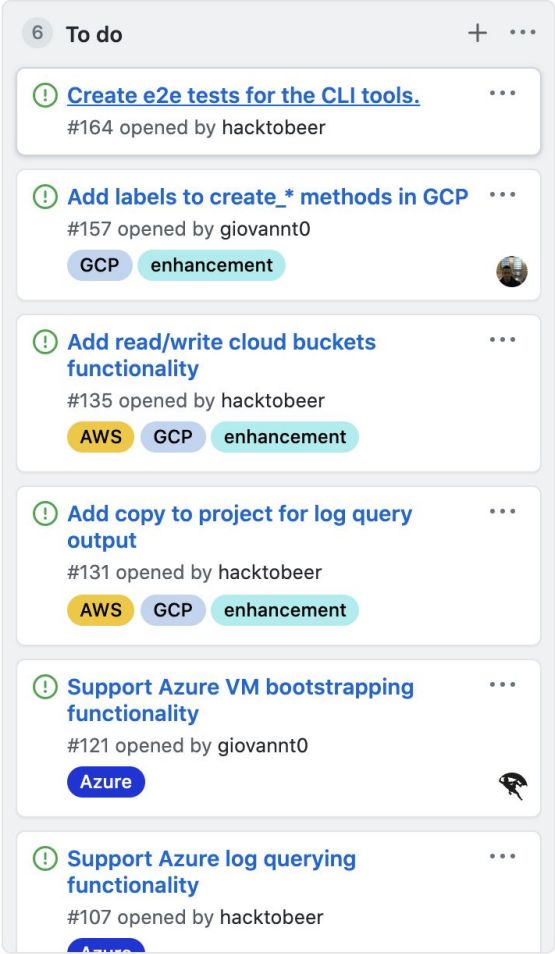
Sticker by Kelly Mahoney @ SoSplush.com
(used with permission)

✨ Code quality ✨

- Type hinting
  - Makes Python a little less YOLO
- Linter checks
  - Helps with code consistency across the entire codebase
- Documentation
  - Verbose docstrings
  - Examples directory

🔥 Issue management 🧯

- “Please open issues”
- Issue labeling
- PRs reference issues
- Public discussion
  - Not easy
  - On issues, PRs comments, etc.
  - https://github.com/open-source-dfir/slack



🚗 Roadmap 🗺️

- GitHub project page
- Horizontal
  - Support basic functionality for more cloud providers (Azure)
  - <Your cloud provider here>
- Vertical
  - More disk operations (instance → dd image)
  - More granular support for logs
- Community
  - More documentation
  - Contributor's guide

https://github.com/google/cloud-forensics-utils/projects

Closing Credits

Links and Contact

https://github.com/google/cloud-forensics-utils

- dfTimewolf
  - https://github.com/log2timeline/dftimewolf
- Turbinia
  - https://github.com/google/turbinia
- Slack Channel
  - https://github.com/open-source-dfir/slack
- Blog
  - https://osdfir.blogspot.com/