Incident Response:
Defending the Gibson in 2015

Darren Bilby - Digital Janitor

dbilby@google.com

ACSC 2015, Canberra

Incidents are Messy

- If it were business as usual you would have stopped it
- Attacker has a plan and they are trying to execute on it
- Need to track, find, and disrupt faster than they can move
- What did they do, why are they doing it, what are they going to do next?
- You probably don't have what you need centralized

Lessons

Gone in 60 seconds

- Every second matters
- Every context switch hurts
- Every handoff to another operator reduces your chance of success

- Empower a small group of people with powerful customizable tools
- Hunch to confirmation in < 30 minutes

The Plan

1. Create an artifact that includes all the logs and files we care about
2. Hunt across our "fleet" for them using GRR
3. Mount the results data using GRR Fuse
4. Process the logs and files using Plaso
5. Put them into a Timesketch visual timeline
6. In <15 minutes

Artifact -> Hunt -> Fuse

Process Log Files

Visualize Global Timeline

Open Source and IR at Google

Forensics and Response Software

- Small market, extremely complex software
- Google is a strange customer - scale, security, platform diversity
- All tools are broken. We can't wait for fixes
- Heavy investment over 5 years
- Everyone needs to level up

GRR RAPID RESPONSE

Enterprise IR

Plaso - Forensic Log Parsing / Timelines

Rekall

Memory Forensics Framework

timesketch

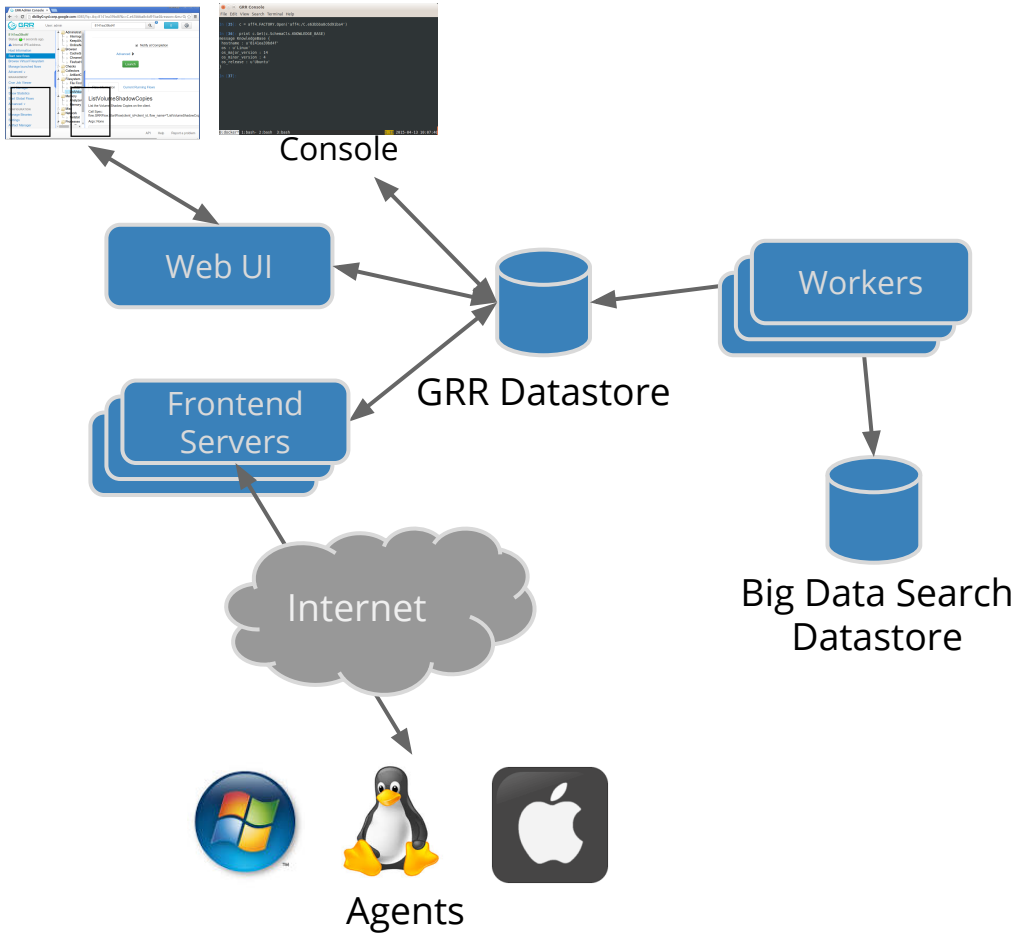
Timeline Visualization

Libyal - Forensic File Format Parsers, Disk Encryption etc

GRR Rapid Response

Agent Based Distributed Forensics and Response

- Built because nothing else worked how we do:
  - 30 analysts at once
  - Scales to 200K + agents
  - Stable, Customizable
  - Mac, Linux, Windows agents
- Google sponsors 4+ full-time staff, other organizations also contribute full-time devs
- Sometimes compared to MIR, Encase Enterprise
- Python, C++, Protobufs
- https://github.com/google/grr


Console
Web UI
Workers
GRR Datastore
Frontend Servers
Internet
Big Data Search Datastore
Agents

GRR Rapid Response

Some Lesser Known Facts

- File and block based deduplication
  - Collecting everything is cheap
- Artifacts are the new IOC (not really)
- Everything is scriptable
- Break-glass method to push custom code
- We have a FUSE layer

Data Flow

Collect Logs & Binaries → Mount from Datastore → Extract timestamps → Ingest to Elastic → View Results

GRR Agents

GRR Datastore

Fuse Mount

Plaso Processing

Elastic Search

TimeSketch

Data Flow

Collect Logs & Binaries →

GRR Agents

GRR Datastore

Mount from Datastore →

Fuse Mount

Extract timestamps →

Plaso Processing

Ingest to Elastic →

Elastic Search

View Results →

TimeSketch

GRR Demo Time

Hunting with Custom Artifacts

http://goo.gl/BCtwtM (https://www.youtube.com/watch?v=JciAp0uB7AY)

Full Demo, which covers the rest of the presentation.

Plaso

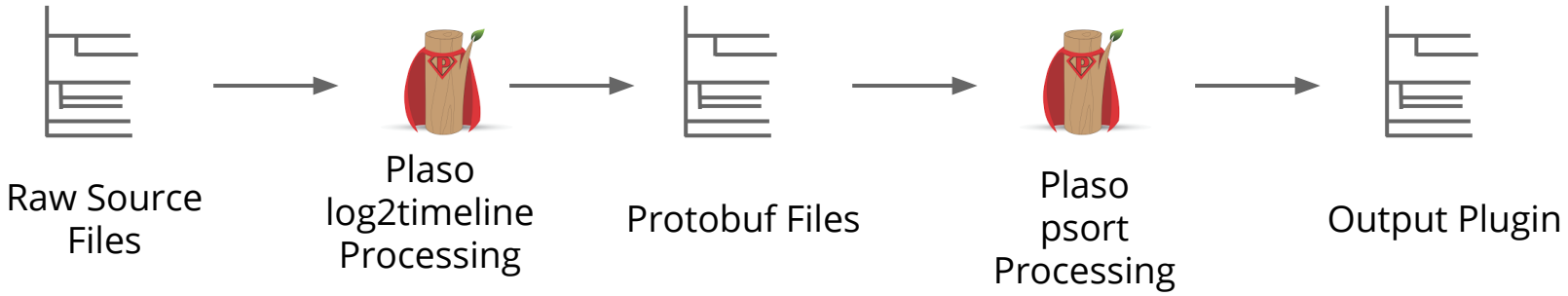
Forensic Timeline Extraction for Everything

- Take a file or filesystem, or set of files and extract all time related information
- Protobuf, Python, C++ libraries
- Easy to customize input, output and parsers
- Lots of external contributors over 5 years (log2timeline)
- http://plaso.kiddaland.net/

Plaso

Forensic Timeline Extraction for Everything

- Allows for bulk processing
- Goto tool for best forensic analysts
- Massive library of parsers
  - Mac, Linux, Windows parsers
  - Handles encrypted images
  - Disks, Registry, Event Logs
  - Browser history, Cache files
  - System restore points
  - ....


Raw Source Files
Plaso log2timeline Processing
Protobuf Files
Plaso psort Processing
Output Plugin

Data Flow

Collect Logs & Binaries → Mount from Datastore → Extract timestamps → Ingest to Elastic → View Results

GRR Agents

GRR Datastore

Fuse Mount

Plaso Processing

Elastic Search

TimeSketch

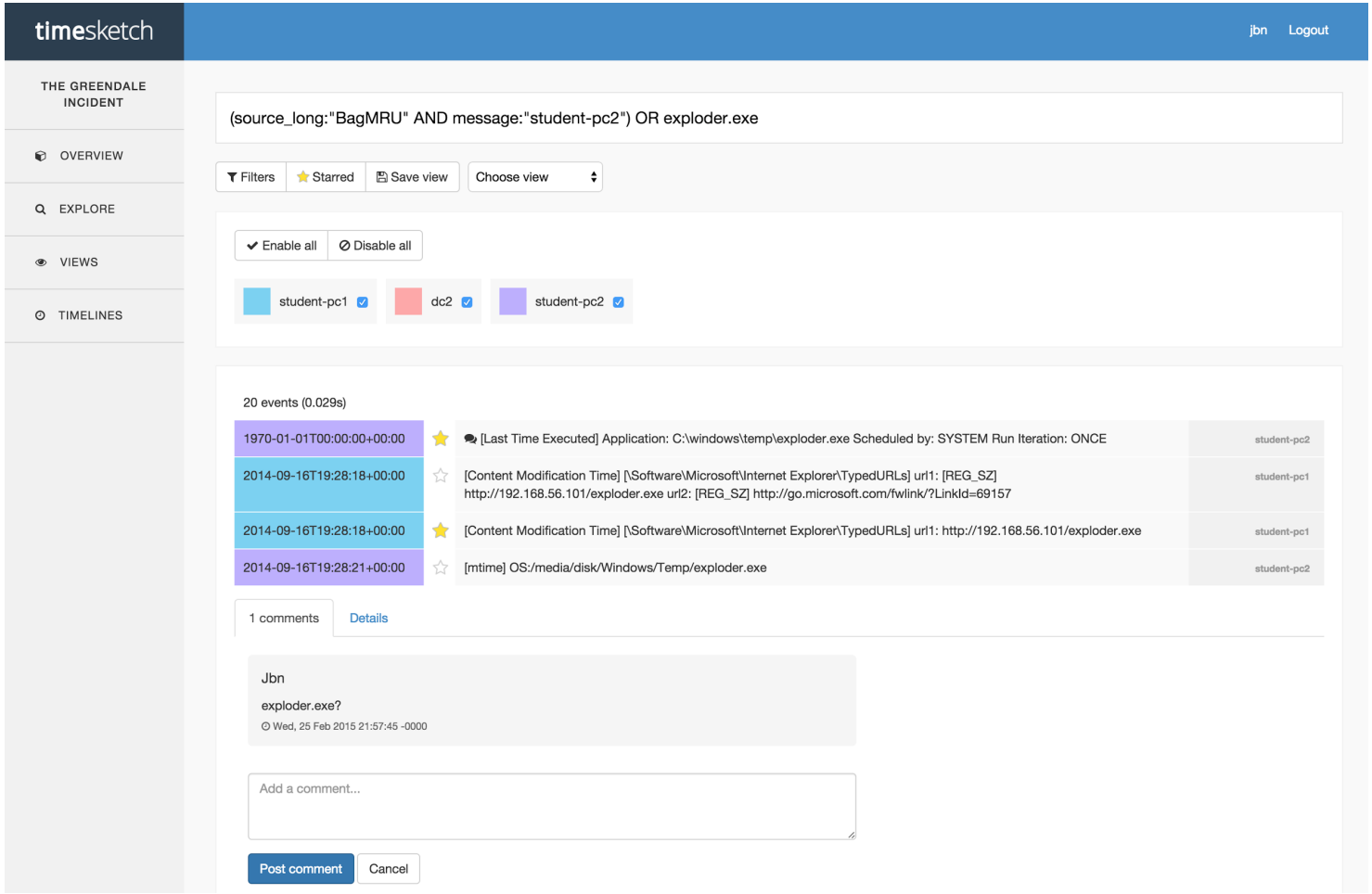
Plaso Processing Demo

Timesketch

Collaborative Timeline Visualization/Filtering/Editing

- New visualization tool for Timeline data
- Fast filtering, annotation
- Collaboration on timeline

- Python + Elasticsearch
- http://www.timesketch.org/
- https://github.com/google/timesketch


timesketch
jbn
Logout
THE GREENDALE INCIDENT
OVERVIEW
EXPLORE
VIEWS
TIMELINES
(source_long:"BagMRU" AND message:"student-pc2") OR exploder.exe
Filters
Starred
Save view
Choose view
Enable all
Disable all
student-pc1
dc2
student-pc2
20 events (0.029s)
1970-01-01T00:00:00+00:00
[Last Time Executed] Application: C:\windows\temp\exploder.exe Scheduled by: SYSTEM Run Iteration: ONCE
student-pc2
2014-09-16T19:28:18+00:00
[Content Modification Time] [\Software\Microsoft\Internet Explorer\TypedURLs] url1: [REG_SZ] http://192.168.56.101/exploder.exe url2: [REG_SZ] http://go.microsoft.com/fwlink/?LinkId=69157
student-pc1
2014-09-16T19:28:18+00:00
[Content Modification Time] [\Software\Microsoft\Internet Explorer\TypedURLs] url1: http://192.168.56.101/exploder.exe
student-pc1
2014-09-16T19:28:21+00:00
[mtime] OS:/media/disk/Windows/Temp/exploder.exe
student-pc2
1 comments
Details
Jbn
exploder.exe?
Wed, 25 Feb 2015 21:57:45 -0000
Add a comment...
Post comment
Cancel

Data Flow

Collect Logs & Binaries → Mount from Datastore → Extract timestamps → Ingest to Elastic → View Results

GRR Agents

GRR Datastore

Fuse Mount

Plaso Processing

Elastic Search

TimeSketch

Everything Else

- Trigger memory collection on NIDS alert

```
flow.GRRFlow.StartFlow(client_id=client_id, flow_name="MemoryCollector",
rdfvalue.MemoryCollectorAction(action_type='DOWNLOAD'))
```

- Push script for quarantining hosts

```
flow.GRRFlow.StartFlow(client_id=client_id, flow_name="ExecutePythonHack",
hack_name='halt_network_hack.py')
```

- Search Memory for Signature

```
flow.GRRFlow.StartFlow(client_id=client_id, flow_name="ScanMemory",
grep=rdfvalue.BareGrepSpec(regex=r'HACK_THE_G.....!')
```

Summary

- Open source tools for IR are extremely capable
- Liberally licensed with Apache - use what you want how you want
- Moving fast means having flexible, fast tools

Questions?

dbilby@google.com

github.com/darrenbilby/grrdemos
docker hub: darrenbilby/grrdemo

github.com/google/grr
github.com/google/timesketch
github.com/ForensicArtifacts
github.com/log2timeline/plaso

google.com/jobs :)