Greg Castle
@mrgcastle

Who am I

GRR Developer, Google IR team

OS X Security

Former lives: pentesting, IR, security audits etc.

Live forensics

GET /beacon HTTP/1.1
Host: evil.com

from Joe’s machine


Joe


GET /beacon HTTP/1.1
Host: evil.com
Joe is on vacation with 3G internet

New APT Report

New malware report BEAR EAGLE SHARK LASER is out: check all the things

New malware report BEAR EAGLE SHARK LASER is out: check all the things

50+ IOCs for Win/Mac and “all the things” is the machines of a highly mobile global organisation with 50k+ employees

GRR: GRR Rapid Response

Open source live forensics

Agent -> Internet -> Server

Disk Forensics = Sleuthkit

Memory Forensics = Rekall

Scalable

Stable, low-impact client

Full-time devs

Why build?

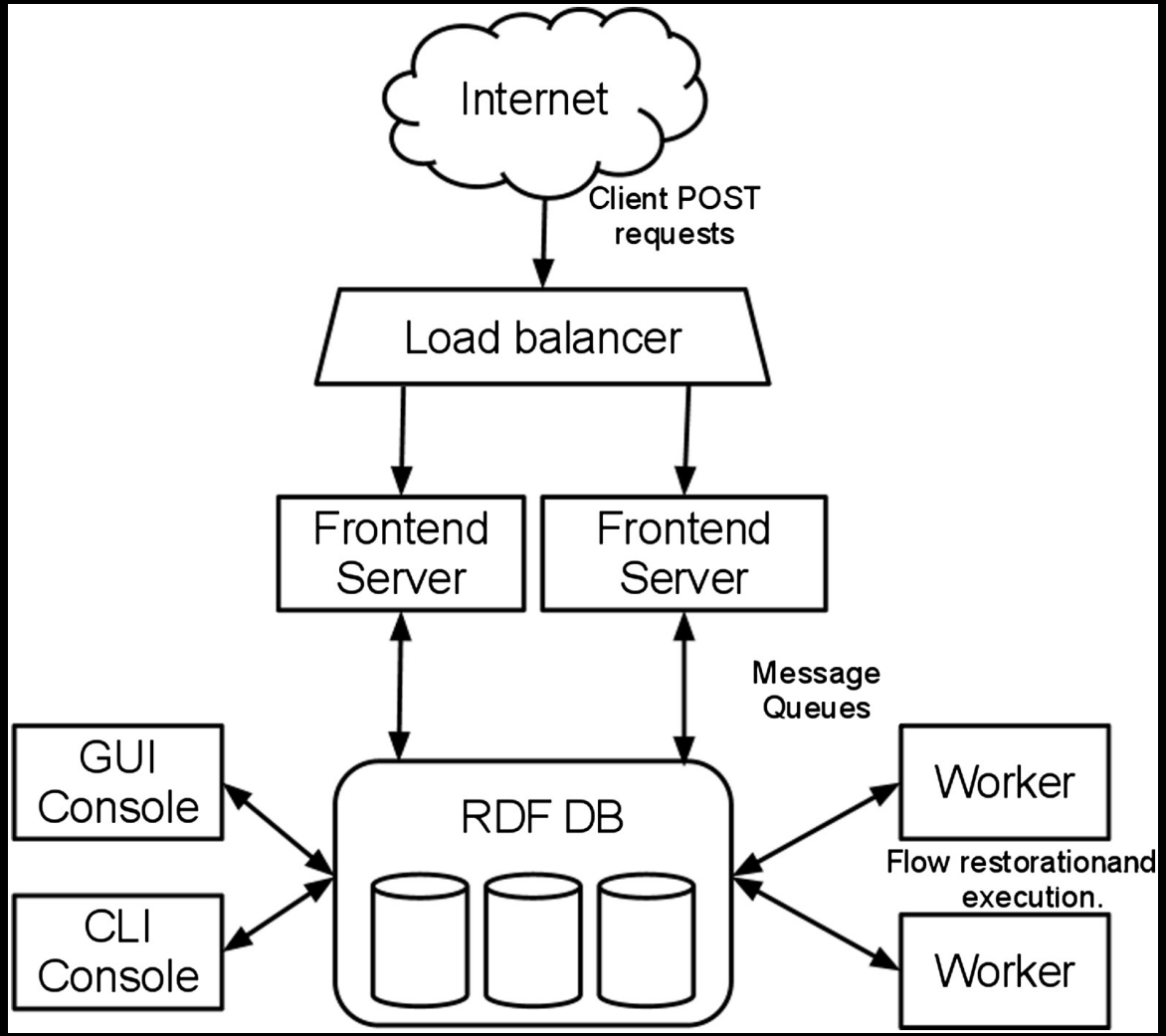
Customize for our threats/detection/defense

50 people analyzing 50 machines

Move as fast or faster than the attacker

Support Mac/Win/Linux


incident response
without remote live forensics


Internet
Client POST requests
Load balancer
Frontend Server
Frontend Server
Message Queues
GUI Console
CLI Console
RDF DB
Worker
Flow restorationand execution.
Worker

Clients

Stable, robust, low impact

Monitored

Limited

10min poll

Smart Server, basic client

Time travel backwards

Faster build/fix/deploy

Less updating

Simpler backwards compatibility

Leak less intent

Server

Frontends pass messages

Workers do the real work

Everything is asynchronous

Queue work on the server

Datastore

Abstracted: easy to switch

MySQL (x2) | MongoDB | SQLite (sharded)

Versioned Data -> axis of time

Demo

Authorization, Auditing

2-party authorization for machine access

DB logging

Audit events

Approval emails with justifications

Fast, reliable, remote.

Advanced live forensics at scale.

Be really really good at collecting

Filesystem/Registry artifacts (Sleuthkit)

Memory artifacts (Rekall)

From difficult-to-specify locations

SANS: “a combination of description, location, and interpretation”

I prefer

“that stuff I want”

As seen in the wild

HardDrive\Documents and Settings\USERNAME\Local Settings\Application Data\Google\Chrome\User Data\Default\History

HKU\S-1-5-21-xxxxxxxxx-xxxxxxxxx-xxxxxxxxx-xxxx\Software\Microsoft\Windows\CurrentVersion\Uninstall\Dropbox\InstallLocation

/Users/<user>/Library/Mail Downloads/

/home/user/.local/share/Trash/

What do I do with these?

HardDrive\Documents and Settings\USERNAME\Local Settings\Application Data\Google\Chrome\User Data\Default\History

HKU\S-1-5-21-xxxxxxxxx-xxxxxxxxx-xxxxxxxxx-xxxx\Software\Microsoft\Windows\CurrentVersion\Uninstall\Dropbox\InstallLocation

/Users/<user>/Library/Mail Downloads/

/home/user/.local/share/Trash/

Common language for interpolation

%%users.localappdata%%\Google\Chrome\User Data*\History

HKEY_USERS**%%users.sid%%**
\Software\Microsoft\Windows\CurrentVersion\Uninstall\Dropbox\InstallLocation

%%users.homedir%%/Library/Mail Downloads/

%%users.homedir%%/.local/share/Trash/

Artifact

```
name: ApplicationEventLog
doc: Windows Application Event log.
collectors:
- collector_type: FILE
  args: {path_list: ['%%environ_systemroot%%\System32\winevt\Logs\AppEvent.evt']}
conditions: [os_major_version >= 6]
labels: [Logs]
supported_os: [Windows]
urls: ['http://www.forensicswiki.org/wiki/Windows_Event_Log_(EVT)']
```

Artifact repository: get it here

~100 artifacts:

github.com/ForensicArtifacts/artifacts

Independent and reusable by any tool

Used and maintained by us

Review, bug reports, patches very welcome

Exporting data for analysis

Heavy data analysis outside GRR

HTTP RPC APIs

Export plugin system:

- CSV
- <elasticsearch or your plugin of choice here>

What’s coming

Event triggered collection

C++ client

More powerful artifact collection

Client m’t: building, reporting, labelling ++

Great, how do I try it?

- Get a 64bit ubuntu machine
- Run the quickstart script from github.com/google/grr
- Open a browser
- Download and install the client on a machine

GRR (and friends) links

github.com/google/grr

github.com/ForensicArtifacts/artifacts

rekall-forensic.com

plaso.kiddaland.net/

github.com/google/timesketch

github.com/libyal/libyal/wiki/Overview