GRR Meetup: API Edition

Nov 2015

Greg Castle @mrgcastle

Mikhail Bushkov

Why have an API?

Made sense architecturally for UI

More important: automation

Why have an API?

Tons of actionable signals boil down to “this file on this machine is weird, look at it”

Eliminate delay: waiting on analyst, waiting on machines to come back online

RESTful API vs. GRR console

Console:

- Too highly privileged for wide use
- Requires shell access to server
- Need to fill out lots of protos

HTTP API:

- Building it anyway for the UI
- Scoped interface, custom ACLs, simple calls

Where are we at

Majority of the UI now uses RESTful API and AngularJS

Writing remote-caller-friendly targets:

- “RemoteGetFile” ready to use

Where are we at: Authorization

ACLs control callers

Require approvals to access content once downloaded

Where are we at: call safety

Sensible limits on file sizes (overridable)

Daily call count limits

No exact dup calls within x minutes

What’s next

Extend API to cover most of GRR functionality

Remove the need to pass authentication token with requests

What’s next: more docs, proto3

Use protobufs v3 to define requests and responses

Autogenerate documentation from protobufs definitions

Make the API as RESTful as possible

What's next: client libraries

protobufs v3: autogenerate client libraries for Python, Java, Go, etc.

Client libs only depend on proto definitions

How others are using it

Up next: @keithtyler expands on his blogpost