The Worst Outage I Never Caused

Julien Goodwin
Google Australia

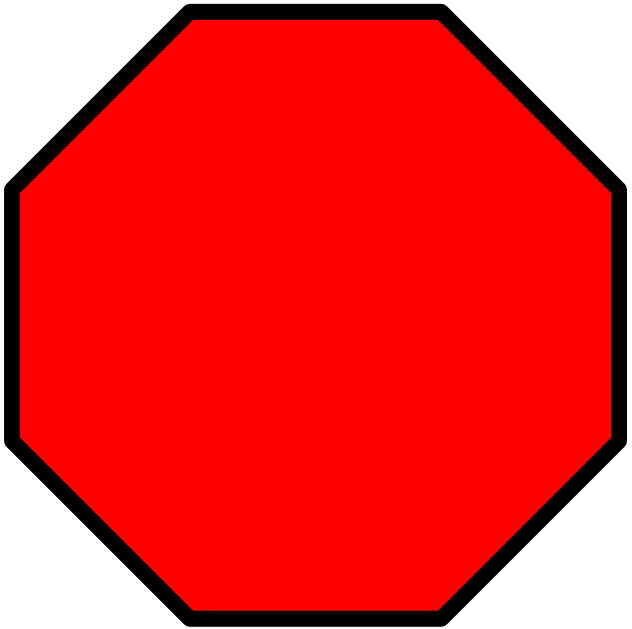
linux.conf.au 2021

Google

Historical Note

This incident happened in April 2017, policies, reactions, safeties, …, have all changed since.

This is much less about any individual element, and more to get you thinking about near-miss events.



Cast of Characters

Julien Goodwin
Network SRE
Sydney, NSW, Australia
UTC+11

Chris Morrow
Network Security
Reston, VA, USA
(Near DC)
UTC -5

The Incident

The ~~Incident~~ Near-Miss

One Monday Morning...

Chris: As it's Monday morning for you, but still Sunday night for me, would you mind rolling that change out?

Julien: Sure

<triggers push tool>

The catch

Luckily the push tool takes a little while to confirm what is in scope.

While it was doing that I went to verify the diff, in case there were other unpushed changes.

Sure enough, rev #5 was live, we're pushing #8.

```
$ p4 diff2 file#5 file#8 | wc -l

<a big number>
```

The Inspection

```
$ p4 diff2 file#5 file#6 | wc -l
```

<small>

```
$ p4 diff2 file#6 file#7 | wc -l
```

<small>

```
$ p4 diff2 file#7 file#8 | wc -l
```

<big> … but that's our simple change, why's it big?

The Inspection - Part 2

```
$ p4 diff2 file#7 file#8
```

So what are we changing?

There's the discussed change, which is small & fine.

We're also removing a magic number from every route … possibly ok, I remembered discussing it, thought we'd do it later.

```
$ grep <magic-number> <magic-number-list>

IMPORTANT_THING: <magic-number>
```

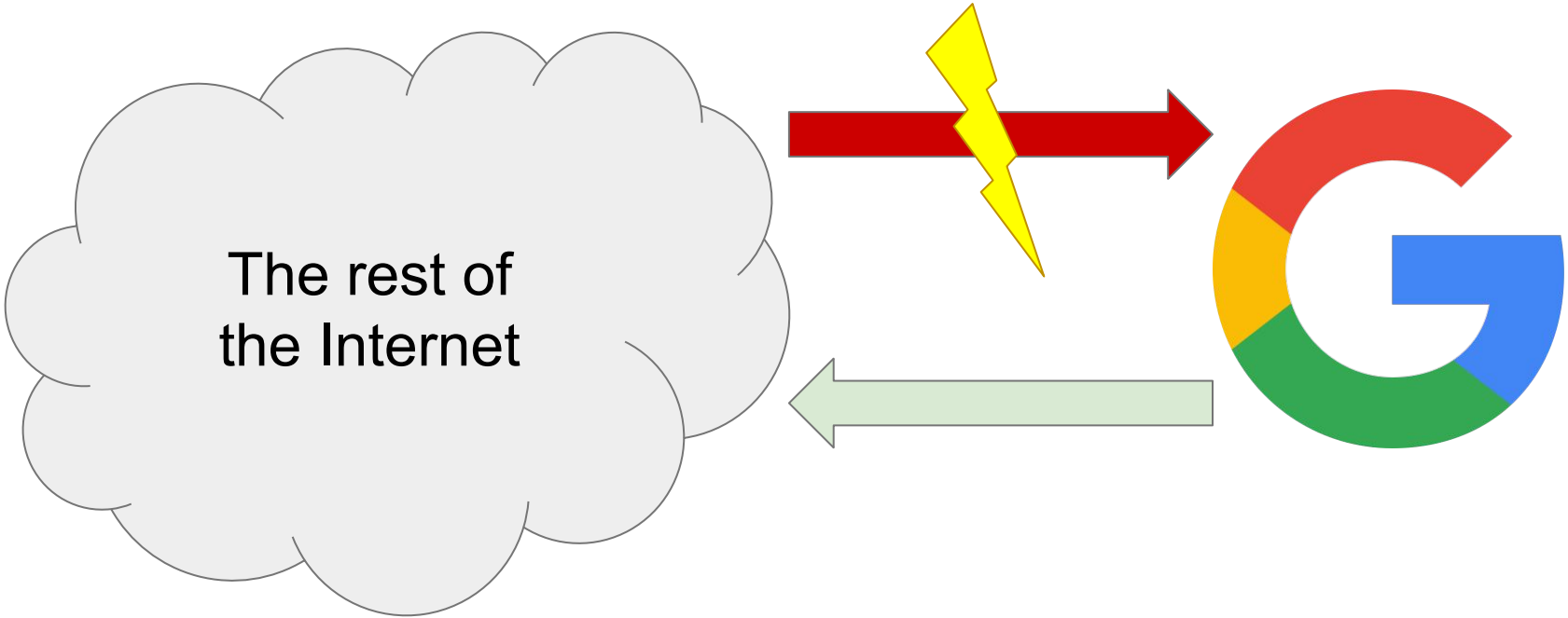
The rollback

… luckily that push tool from earlier requires a final approval after confirming exactly what devices will be modified.

Ctrl-C took care if it.

A quick revert changelist of the file removed the danger.

The potential impact



The build up

Sources of routes on the Internet

Directly advertised by a router connected to a LAN.

Dynamic advertisement from systems like load balancers.

Aggregating the above.

Static Routes.

What do static routes look like?

```
/* Google Public DNS */
route 8.8.8.0/24 {
    community [ 15169:10100 15169:10110 15169:10120
15169:10130 15169:10140 15169:10150 15169:10160 15169:10200
15169:20210 15169:20220 15169:20221 15169:20300 15169:20310
15169:30320 15169:30330 ... ];
    next-hop discard;
}
```

Lots of magic numbers!

- `15169:10100` — Advertise to the Internet
- `15169:10110` — Advertise to Peers
- `15169:10120` — Advertise to Internet Exchanges
- `15169:20480` — Prepend route
- `[15169:10100 15169:20480]` — Do magic behavior

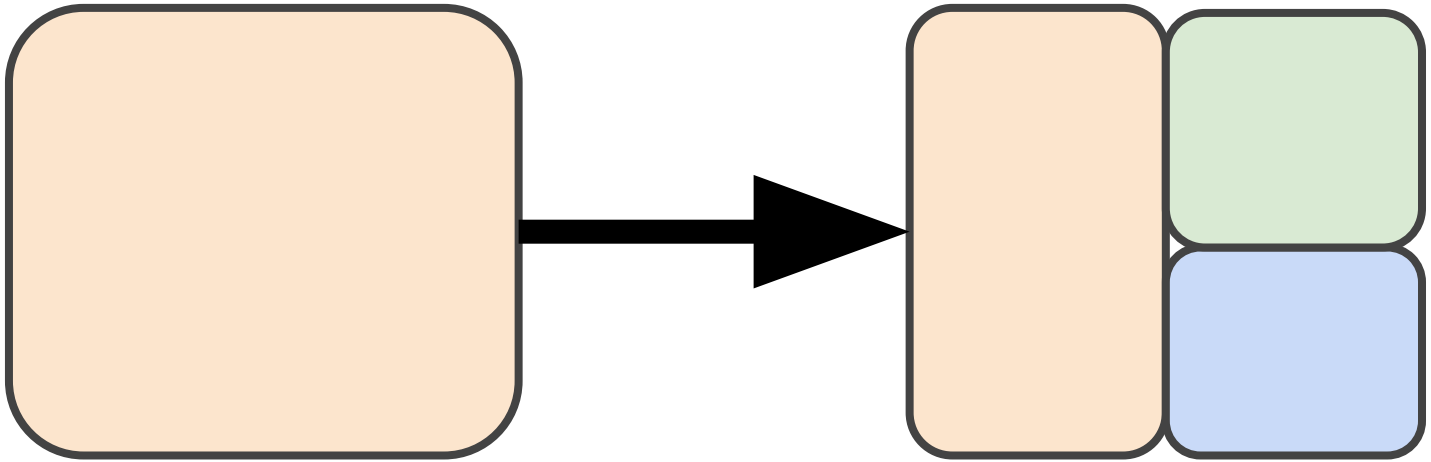
Unhelpful tooling

```
$ explain-magic-number 15169:10100

15169:10100 MAGIC-BEHAVIOR
```

The Intended Change

Break up a big IPv4 address block into smaller blocks with distinct purposes.



The Change Review Discussion

Thursday Morning: (SYD time)
Julien: We're not advertising these routes externally, so let's not set those communities.

<... some discussion ...>

Saturday Morning:
Chris: We also have these extra communities including "Do magic behavior", should they go to?
Julien: Yeah, that can go from everywhere, it no longer does anything
Chris: Done
<Change submitted>

The root causes

Root causes

- Obsolete config not fully cleaned up
- Magic numbers in places humans need to deal with
- (Almost) no tests
  - One simple one validated syntax, but not content
- No simulation
- No clear ownership
  - Routing configuration regularly changed by people across three disparate teams, none of which "own" it.
- In short, classic haunted graveyard

Actions taken

Quiet whistling...

Nobody notices a near miss, maybe I can just ignore it and move on.

Quiet whistling...

Nobody notices a near miss, maybe I can just ignore it and move on.

twitch

Would having the outage have been better?

Trade a major press-worthy outage for more effort to actually fix things?

I wrote a Post-mortem

… three days later I was still twitching.

Very few people cared.

Since this was a near miss, despite the potential impact, couldn't get engagement on it.

Had the incident happened would certainly have had many meetings with VPs etc.

Still, a great opportunity to document the state of key network elements.

Tests!

Used an existing parser library for Juniper-style configuration I wrote some basic tests in Python.

- Each entry is "as expected"
  - Standard text is exactly the standard text
- At least N routes of each major type
- Uniqueness
- The communities are only from the limited expected set
- Topology tests
  - Things expected on top level routes are there, and only there
- RFC1918 (et al) space not trying to be advertised to the Internet

Config Generator

A few months later I wrote an interim Python generator that took a more human readable (text protocol buffer) input and generated the config file as output.

Along with a test to ensure the two stayed in sync.

```
route {
  prefix: "8.8.8.0/24"
  name: "Google Public DNS"
  community: "AS15169.EXT"
  community_set: "GLOBAL"
}
```

Questions?