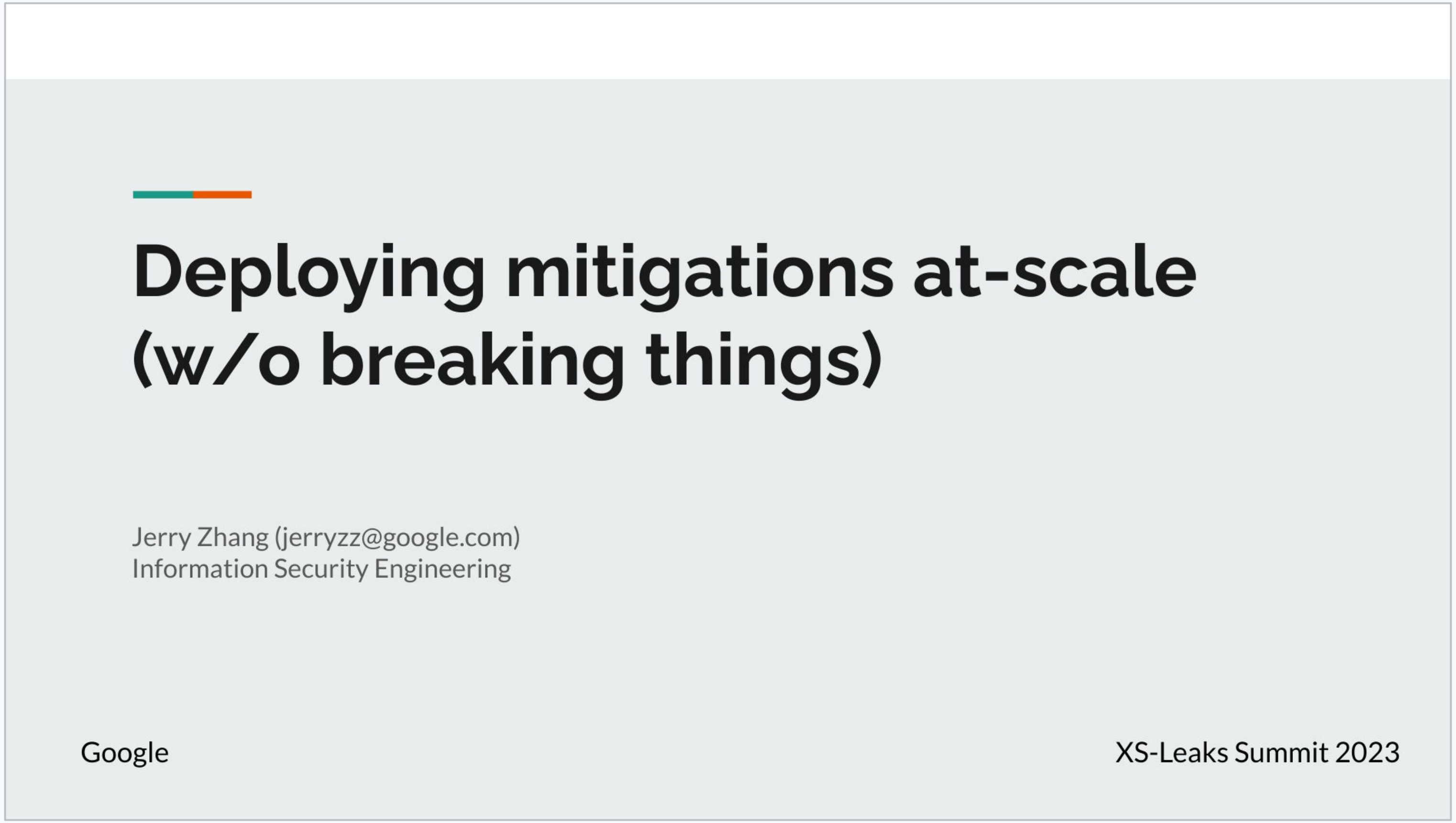
XS-Leak Protections at Google

David Dworken
ddworken@google.com

Sep 2024

In 2023:



Deploying mitigations at-scale (w/o breaking things)

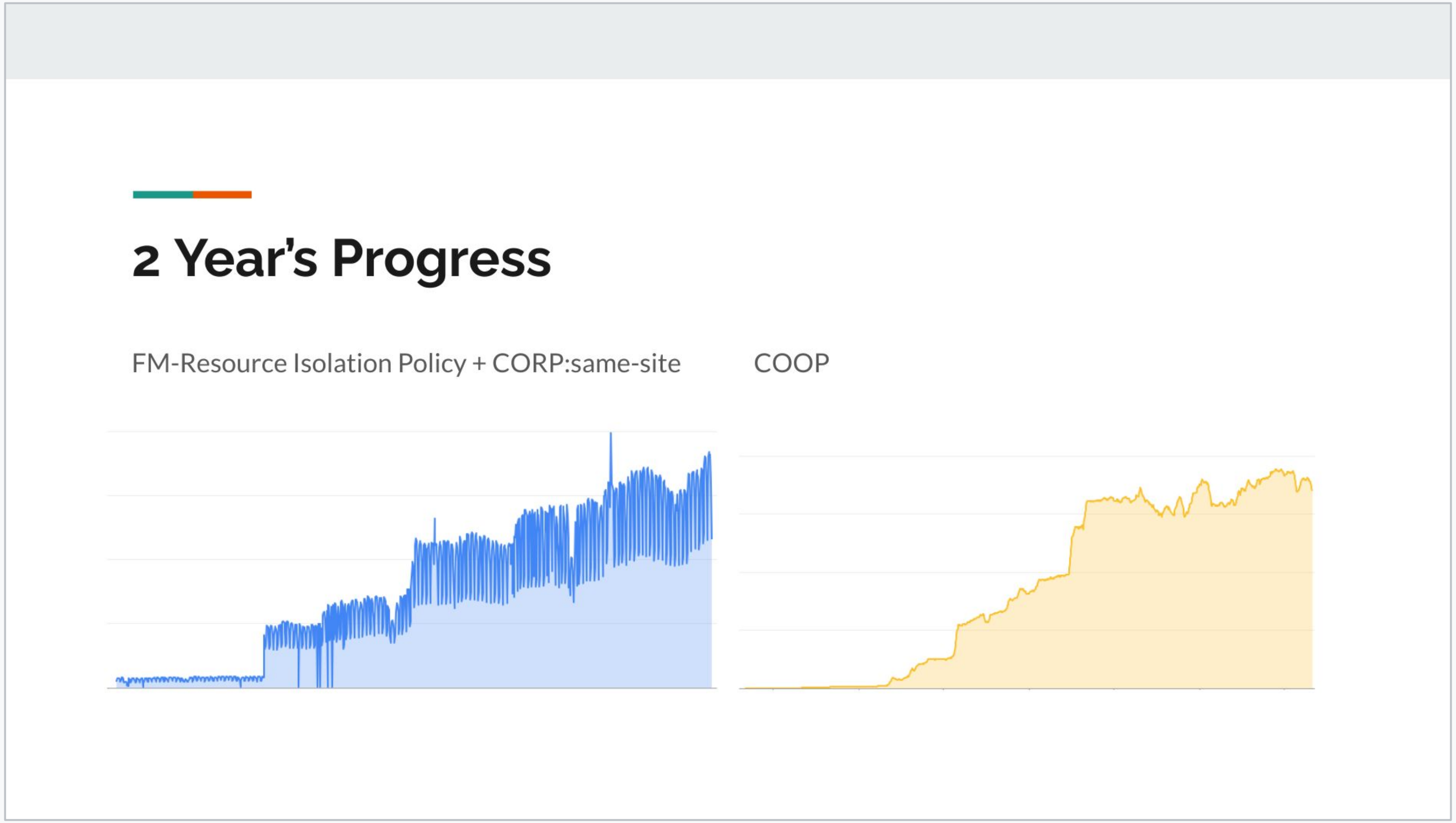
Jerry Zhang (jerryzz@google.com)
Information Security Engineering

Google

XS-Leaks Summit 2023

Google

In 2023:

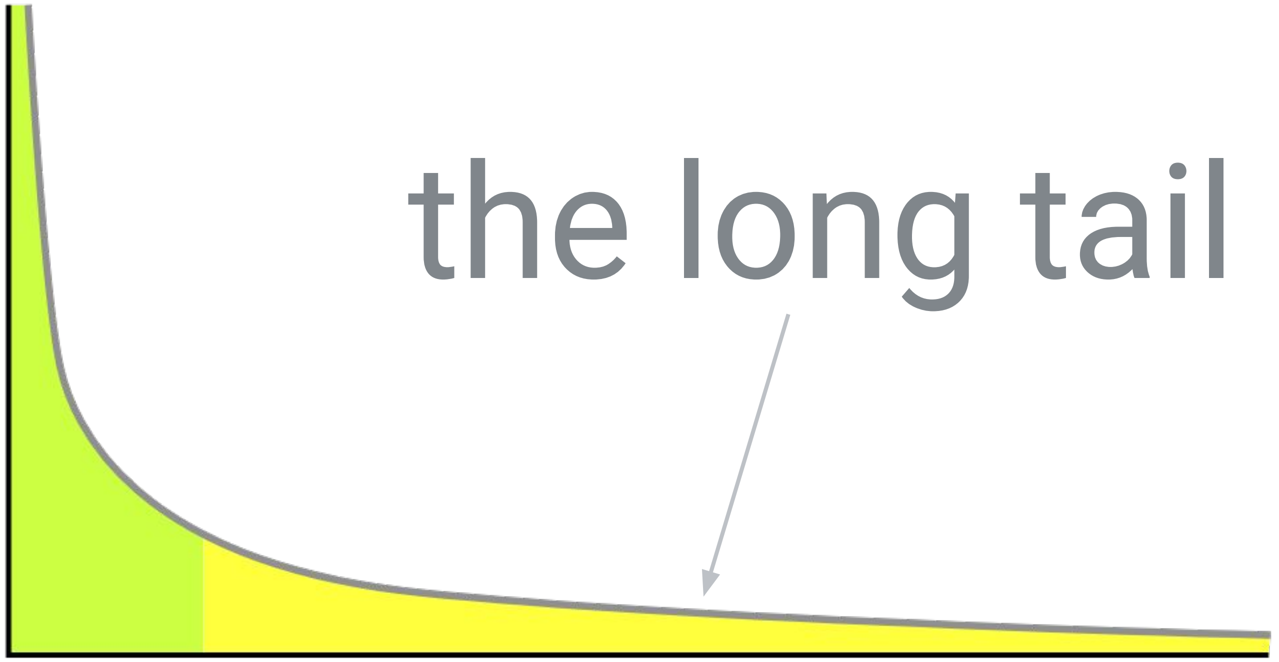


2 Year's Progress

FM-Resource Isolation Policy + CORP:same-site

COOP

For a few years now, we've been making progress on XS-Leak protections... *so now what?*



Source: https://en.wikipedia.org/wiki/Long_tail#/media/File:Long_tail.svg

Frontier #1: Non-Standard Frameworks

Google has a massive ecosystem of web services and there is a long-tail of services built on non-standard web frameworks

- New services are built on the well-lit path that has had XS-Leak protections built in for the past few years
- Many Google services have existed for decades and might be built in non-standard ways (e.g. homegrown C++ frameworks)

We need to ensure that bespoke frameworks are secure

Frontier #1: Non-Standard Frameworks

New Strategy: **Sandboxing**

We're running large rollouts to ensure that services built on non-standard frameworks are *sandboxed*:

- Mitigate isolation vulnerabilities (including XS-Leaks) by enforcing:
  - Resource Isolation Policy (RIP) and Framing Isolation Policy (FIP) to block cross-site requests
  - Cross-Origin Opener Policy (COOP) to restrict popups
- Mitigate injection vulnerabilities by enforcing CSP sandbox to isolate injection vulnerabilities into a `null` origin

Frontier #1: Non-Standard Frameworks

But there's a contradiction, if a page sets CSP sandbox:

- It will execute in a `null` origin
- So requests it make will be `Sec-Fetch-Site: cross-site`
- So RIP/FIP will block any requests that the page makes 😱

This means we can't deploy CSP sandbox and FM isolation together!

Solutions:

- [For now] Deploy them one at a time and use report-only mode to ensure we don't break anything
- [Long term] Browser-level discussions about:
  - `allow-unique-origins` to allow sandboxed pages to execute in a non-`null` origin
    - https://github.com/WICG/proposals/issues/121
  - Adding an opt-in CSP sandbox flag to change the cross-site behavior of requests
    - https://github.com/w3c/webappsec-csp/issues/664

Frontier #2: COOP Gaps

Sometimes deploying COOP is ***really*** hard

Frontier #2: COOP Gaps

How can we make COOP easier to deploy?

- COOP restrict-properties

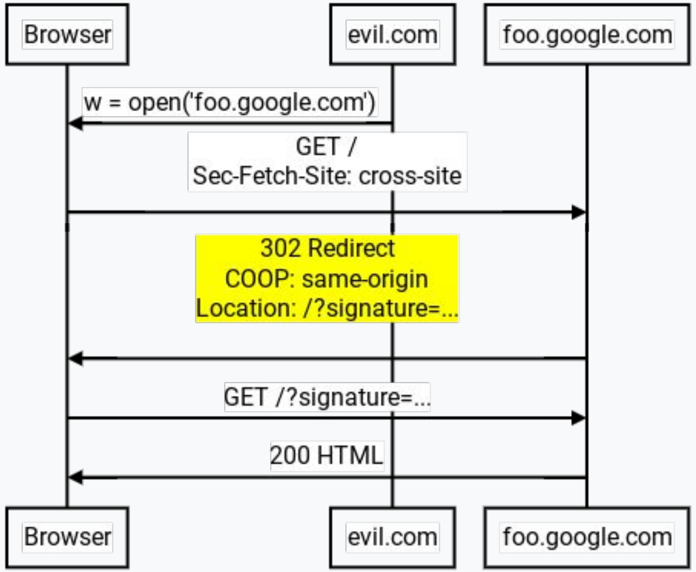
Frontier #2: COOP Gaps

How can we make COOP easier to deploy?

- ~~COOP restrict-properties~~
- Not deploying COOP at all

By intercepting cross-site requests and serving a redirect with COOP, **we break the window connection without having to turn on COOP for the main application**

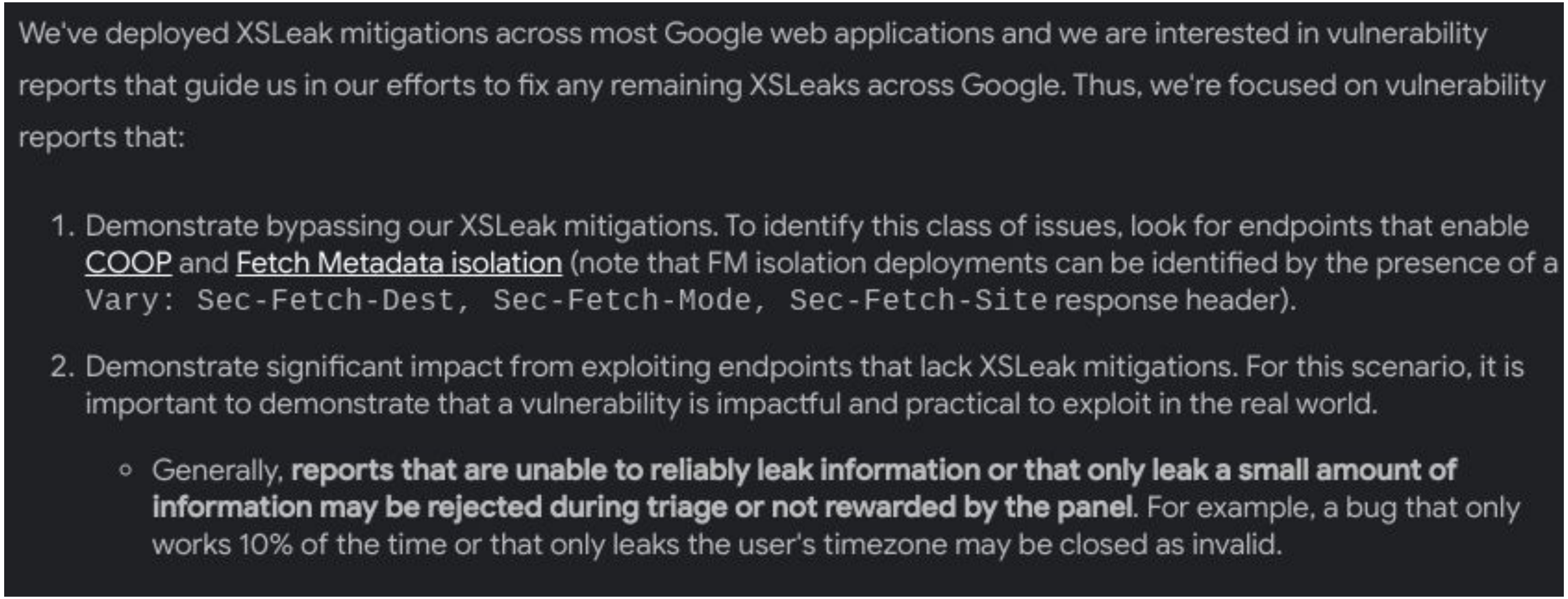
- Helps in certain cases where turning on COOP is prohibitively difficult



⚠️ Warning: Experimental Approach ⚠️

Thanks to these tricks and continual effort, XS-Leak mitigation coverage is always increasing!

XS-Leak Vulns are officially back in-scope for the VRP!



We've deployed XSLeak mitigations across most Google web applications and we are interested in vulnerability reports that guide us in our efforts to fix any remaining XSLeaks across Google. Thus, we're focused on vulnerability reports that:

1. Demonstrate bypassing our XSLeak mitigations. To identify this class of issues, look for endpoints that enable COOP and Fetch Metadata isolation (note that FM isolation deployments can be identified by the presence of a `Vary: Sec-Fetch-Dest, Sec-Fetch-Mode, Sec-Fetch-Site` response header).
2. Demonstrate significant impact from exploiting endpoints that lack XSLeak mitigations. For this scenario, it is important to demonstrate that a vulnerability is impactful and practical to exploit in the real world.
   - Generally, **reports that are unable to reliably leak information or that only leak a small amount of information may be rejected during triage or not rewarded by the panel**. For example, a bug that only works 10% of the time or that only leaks the user's timezone may be closed as invalid.

https://bughunters.google.com/about/rules/google-friends/6625378258649088/google-and-alphabet-vulnerability-reward-program-vrp-rules
https://bughunters.google.com/learn/invalid-reports/web-platform/xsleaks/5022006283862016/xsleaks-and-xs-search

Questions?