envoycon

Hardening Envoy

Alyssa Wilk
Senior Staff Engineer @Google

Hardening Envoy

- Handling new use cases
  - Envoy as an unprotected edge proxy
  - Envoy as a cloud proxy

- Software improvements
  - Resilience against external attacks
  - Resilience against (accidental) internal attacks
- Test improvements
  - Integration tests
  - CI (Continuous integration)
  - Fuzzing
  - Loadtesting

Resilience against external attacks

envoycon

Flow Control (#150)

- Closes the trickle-attack vector
  - Fast upstream and slow downstream leading to OOM
  - Botnet + few resources: global outage

- In-memory buffer limits on every buffer from the Network::Connection to individual filters
- As buffers fill up, they signal the source of incoming data to back off, via TCP congestion control or H2 flow control

Resilience against external attacks

Circuit breaking (#373)

- Closes two other fun DoS vectors
  - Keep opening new streams until OOM
  - Keep opening new connections until fd crash
  - High potential for cascade failure
  - Especially dangerous when coupled with prior lack of keep-alive timeouts (#3841) and connection leakage (#3813)

- On approaching fd limits, stop accepting new connections.
- On approaching memory limits, stop accepting new streams.

With flow control, provides upper bound on system resources.

Resilience against internal attacks

- Enhanced load balancing
  - Priorities
  - Improved fairness
  - Flexibility
- Configuration safeguards
  - Proto validation to avoid poison configs
  - Reload and roll-back tests (coming soon!)

Integration test improvements

- 90% lower LoC for testing happy path
- Utilities to test end to end configuration
- Automated upstream fuzzing timing conditions
- Easier cross-protocol testing
- Improved debugability
- Coming soon: configuration reload framework

CI improvements

- asan / tsan / ubsan / clang-tidy
  - Regularly catch issues before they’re merged
- Increased CI runs catch real production bugs
  - Backup bug (early responses causing 500s)
  - Connection leak (connection resets being missed)
- Detailed instructions for reproducing and deflaking test failures

Fuzzing

- 14 fuzzers covering untrusted code (e.g. http, h2, …) as well as internal (config fuzzing)
- Run in CI, Cluster-fuzz
- 80+ PRs, 35+ bugs fixed: #4814 #4751 #4737 #4731#4576 #4378 #4377 #4346 #4328 #4321 #4313 #4307...
- Bugs which could crash Envoy, cause deadlock, fail internal code assertions, allow buffer overflow

Loadtesting (coming soon)

- Work in progress by WeAmp
- Catch bugs before they hit prod, for Lyft or for any other users
- In the last 6 months: #4382 #4276 #4295 #4043 #3609 #3590
  - Combination of crashes, CPU regressions, and straight up buggy functional changes
  - Doesn’t even include ‘found bugs’ more than a week or two later which would more than double the number.

Adoption!

- While production is the last place you want to catch issues, it's bound to happen

- Envoy now is used by dozens of companies with tens of thousands of instances, tens of millions of QPS.
  - From graduation proposal: Lyft, Cookpad, AppDirect, Pinterest, Coursera, Salesforce, GO-JEK
  - From Envoycon: Google, Stripe, Square, Alibaba, eBay, Yelp, Covalent, Groupon, Voltera, Stripe, Microsoft,
  - And many more!
- Different deployment patterns catch different bugs - the more users Envoy has the faster issues will get found.

Questions?