Formal Methods at Scale

Organizer and scribe takeaways from a talk given by
Byron Cook (Amazon Web Services) at the
FM@Scale West Meeting on October 9, 2019

As the world's most comprehensive and broadly adopted cloud services providers, AWS has long made security its highest priority. In addition to an abundance of security resources and expert guidance, AWS is applying automated reasoning at scale to raise the level of assurance of their foundations (e.g. cryptography, virtualization, storage), and also to help customers help themselves in the fight against security concerns such as data breaches.

Byron Cook, senior principal scientist at AWS, spoke at the FM@Scale in Palo Alto, where he provided detail on the array of automated reasoning activities at AWS. Specifically, Byron spoke about:

- Reasoning about customer-authored identity and access management configurations using SMT solvers as seen in service features such as IAM Access Analyzer, S3 Block Public Access, and Config Rules.
- Reasoning about customer-authored virtual networks using SMT solvers, as seen in Amazon's Inspector service
- Mechnically proving correctness properties of low-level implementation code, e.g. the code that implements Amazon's cryptography, virtualization, and storage infrastructure.
- Mechanically proving the correctness of protocols that power the cloud, e.g. the security of Amazon's distributed secrets protocol behind the Key Management Store (KMS) service, or the durability of the sharding protocol that powers Amazon's S3.

Byron's talk addressed some of the open pain-points that his team and AWS as a whole faces as they attempt to further scale their efforts. Specifically:

1. Connecting proofs about models of systems and protocols to the actual code. Current tools such as TLA+ allow teams to model their designs, but do not facilitate CI/CD integrations later. CI/CD, it turns out, is the most valuable aspect of formal verification over time.
2. Connecting threat models and mitigation plans developed during proactive pre-launch security reviews to formal properties
3. Proving the soundness of the SMT solvers, and tracking/reporting the environment assumptions made during mechanical proofs
4. Seemless integration with CI/CD systems, including the (semi-automated) "repair" and long-term maintenance of proofs in both models & code
5. Auditable proof production for some tools and static analysis algorithms, e.g. abstract interpretation. Here the application is regulatory compliance, where proofs can be used as artifacts for audits that are automatically constructed.
6. Usability of the more advanced tools. Coq or even Dafny have steep learning curves. This limits adoption to insiders with PhDs, whereas for scale Amazon is asking recently graduated undergrads to both develop code and prove it using formal reasoning tools
7. Workforce training. As machine learning has suffered in the past, the talent pool for subject-matter experts in automated reasoning is very small. This is limited Amazon's ability to scale efforts out amongst multiple teams.
8. Reasoning about complex concurrent systems, and large large code basis.
9. 9. Support for new and very popular languages, e.g. Python, Rust