One-click formal methods

Liana Hadarean
Amazon Web Services
hadarean@amazon.com

Formal methods have been successfully applied in domains such as microprocessor hardware design and aerospace, e.g., proofs of security properties for helicopter control systems [4]. However, despite 50 years of research and development, we have not seen wide adoption of formal methods for large and complex systems such as web services, industrial automation, or enterprise support software. One of the key difficulties when proving security, safety, and robustness of these systems is the problem of finding the models of system architectures necessary for analysis. Additionally, the size of the potential user community and the business value typically does not justify the creation of scalable and easy-to-use tools for formal verification.

With the cloud, much of this has changed. Descriptions of cloud services provide accurate models in the form of computer-readable contracts. These contracts establish and govern how the system behaves and in many cases these models are amenable to formal analysis at scale. [1] Most importantly, since those models are used by a large user community, it is now economically feasible to build the tools needed to verify those models.

In this talk, we discuss the trend of constructing practical and scalable cloud-based formal methods and how they can easily be used by customers – sometimes with just one-click. [5] At Amazon Web Services (AWS), we have used cloud models to construct large-scale automated reasoning tools that can prove whether or not access controls meet governance rules and whether networks are properly secured. These tools are used millions of times daily and have a direct positive impact on AWS customers.

At AWS, we have developed the Zelkova policy analysis engine [2] to prove properties of access control policies. Zelkova encodes access control policies and properties into the logic of Satisfiability Modulo Theories (SMT). To support millions of queries a day by both internal and external customers, 99% of all Zelkova proofs must complete in 160 milliseconds or less. Zelkova is currently integrated within several AWS public facing services including Amazon S3, AWS Config, AWS IoT Device Defender, Amazon Macie, AWS Trusted Advisor, and Amazon GuardDuty. External customers ranging from the financial industry to compliance regulators use Zelkova to ensure that their access control policies are compliant with corporate governance rules.

Formal methods in the cloud are used for more than just access control. Tiros [3], part of the Amazon Inspector service, uses the model provided by Elastic Compute Cloud (EC2) network configurations to perform proofs of network reachability without generating any network traffic. For example, Tiros can check whether there exists any public IP address on the Internet that can access a local database server. Unlike packet scanning approaches, Tiros will find any such access path, and does not add load to the network.

The AWS Identity and Access Management (IAM) Access Analyzer is a new feature that makes it simple for security teams and administrators to check that their policies provide only the intended access to resources. While Tiros, and Zelkova provide a yes/no answer, IAM Access Analyzer uses formal methods to give insight into who can access what resource. Access Analyzer presents customers with a concise but comprehensive view of who outside their account can access what resources in the form of a list of findings. Using Zelkova as the underlying technology, Access Analyzer can evaluate hundreds or even thousands of policies across a customer's environment in seconds. IAM Access Analyzer can be enabled with just one click across an entire account to continuously analyze permissions granted using policies associated with their Amazon S3 buckets, AWS KMS keys, Amazon SQS queues, AWS IAM roles, and AWS Lambda functions.

Formal methods tools like Zelkova, Tiros and Access Analyzer show we can now use automated reasoning to provide inexpensive and provable assurance to customers. We expect that this trend of building practical and scalable formal methods in the cloud will lead to environments where security, compliance, availability, durability, and safety properties can be proved about large-scale systems.

References

[1] B. Cook, "Formal reasoning about the security of Amazon Web Services", Federated Logic Conference (FLoC) 2018, 2018.

[2] J. Backes, P. Bolignano, B. Cook, C. Dodge, A. Gacek, K. Luckow, N. Rungta, O. Tkachuk, C. Varming, "Semantic-based Automated Reasoning for AWS Access Policies using SMT", in Formal Methods in Computer-Aided Design (FMCAD), 2018.

[3] J. Backes, S. Bayless, B. Cook, C. Dodge, A. Gacek, A.J. Hu, T.Kahsai, B. Kocik, E. Kotelnikov, J. Kukovec, S. McLaughlin, J. Reed, N. Rungta, J. Sizemore, M. Stalzer, P. Srinivasan, P. Suboti, C.Varming, B. Whaley, "Reachability Analysis for AWS Networks", in Computer Aided Verification (CAV) 2019, July 2019.

[4] D. Cofer et al, "A formal approach to constructing air vehicle software", IEEE Software, vol. 51 no. 11, pp 14-23, 2018

[5] J. Backes, P. Bolignano, B. Cook, A. Gacek, L. Kasper, R. Neha et al, "One-Click Formal Methods", IEEE Software, vol. 36, no. 6, pp 61-65, 2019