Status Report: Formal Analysis of Web Security

Karthikeyan Bhargavan[1], Abhishek Bichhawat[2], Quoc Huy Do[3], Daniel Fett[3], Ralf Küsters[3], and Guido Schmitz[3]
[1] INRIA, France
karthikeyan.bhargavan@inria.fr
[2] Saarland University, Germany
bichhawat@cs.uni-saarland.de
[3] University of Stuttgart, Germany
{quoc-huy.do,daniel.fett,ralf.kuesters,guido.schmitz}@sec.uni-stuttgart.de

Many corporate and end-user single sign-on solutions are based on OAuth and OpenID Connect. For a secure operation, both frameworks rely on web technologies with intricate security properties.

In the past, security requirements of such frameworks and web protocols were often under-defined and assumptions were implicit. In the end, security was assessed in an adhoc way instead of in a rigorous and systematic manner. This has led to a number of critical security problems.

Formal methods enable a precise definition and verification of security properties. In previous work, we have developed models of the web that capture important security properties and have successfully applied these models to find new attacks on and verify the security of OAuth and OpenID Connect:

- The WebSpi approach in [1] is based on the applied pi-calculus and ProVerif [3] and aims to make the discovery of security vulnerabilities systematic and partially automated. To this end, some important features of the web infrastructure had to be abstracted away.
- The FKS model [4] used in [5], [6] is the most detailed model of the web infrastructure to date but does not feature automation or tool-support so far. Instead, analysis in the model relies on laborious pen-and-paper proofs.

We aim to combine the comprehensiveness of the FKS model with mechanized (i.e., tool-supported and tool-verifiable) proofs using F* [7] to encode and verify the FKS model and OAuth/OpenID Connect models.

F* is a functional programming language aimed at program verification. Program specifications, including correctness and security properties, can be expressed precisely and compactly thanks to F*'s type system, F* makes use of the SMT (Satisfiable Modulo Theory) solver Z3 as the backend in order to prove that programs meet their specifications. F* can be used to model and verify software and protocols that are subject to critical security requirements: cryptographic constructions and protocols, web browsers etc. It has been applied successfully to verify cryptographic protocols TLS 1.2 and TLS 1.3 [2] including the underlying cryptographic primitives.

With mechanized proofs in the F* web model it becomes easier to reuse proofs. For example, proofs for OAuth could be reused in proofs for OpenID Connect, and changes/extensions to the protocols could be tracked much faster. In a similar manner, regression tests could be run against extensions of the models.

Additionally, the F* specification of a protocol could be used to extract runnable code in JavaScript, TypeScript, OCaml, F# or C for execution. That way, we would be able to build an executable OAuth library from verified code.

If this paper is accepted, we would report on our research agenda and give a short status report of our work (roughly 15 minutes).

References

[1] C. Bansal, K. Bhargavan, A. Delignat-Lavaud, and S. Maffeis. Discovering Concrete Attacks on Website Authorization by Formal Analysis. *Journal of Computer Security*, 22(4):601–657, 2014.

[2] K. Bhargavan, B. Bond, A. Delignat-Lavaud, C. Fournet, C. Hawblitzel, C. Hritcu, S. Ishtiaq, M. Kohlweiss, R. Leino, J. Lorch, K. Maillard, J. Pang, B. Parno, J. Protzenko, T. Ramananandro, A. Rane, A. Rastogi, N. Swamy, L. Thompson, P. Wang, S. Zanella-Béguelin, and J.-K. Zinzindohoué. Everest: Towards a verified, drop-in replacement of HTTPS. In *2nd Summit on Advances in Programming Languages*, May 2017.

[3] B. Blanchet. Automatic Verification of Security Protocols in the Symbolic Model: The Verifier ProVerif. In *Foundations of Security Analysis and Design VII - FOSAD 2012/2013 Tutorial Lectures*, volume 8604 of *Lecture Notes in Computer Science*, pages 54–87. Springer, 2013.

[4] D. Fett, R. Küsters, and G. Schmitz. An Expressive Model for the Web Infrastructure: Definition and Application to the BrowserID SSO System. In *35th IEEE Symposium on Security and Privacy (S&P 2014)*, pages 673–688. IEEE Computer Society, 2014.

[5] D. Fett, R. Küsters, and G. Schmitz. A Comprehensive Formal Security Analysis of OAuth 2.0. In *Proceedings of the 23nd ACM SIGSAC Conference on Computer and Communications Security (CCS 2016)*, pages 1204–1215. ACM, 2016.

[6] D. Fett, R. Küsters, and G. Schmitz. The Web SSO Standard OpenID Connect: In-Depth Formal Security Analysis and Security Guidelines. In *IEEE 30th Computer Security Foundations Symposium (CSF 2017)*. IEEE Computer Society, 2017.

[7] F*: A Higher-Order Effectful Language Designed for Program Verification. https://www.fstar-lang.org. Accessed: 2018-01-18.