Safe Haskell

David Terei[1], Simon Marlow[2], Simon Peyton Jones[2], David Mazières[1]
[1]Stanford University, [2]Microsoft Research

ABSTRACT

Safe Haskell [1] is a new extension to the Haskell language that is implemented in the Glasgow Haskell Compiler (GHC) as of version 7.2. It allows for unsafe code to be securely included into a trusted code base by restricting what features of the language code is allowed to access. Put simply, it makes the types of programs trustable. Safe Haskell itself is aimed to be as minimal as possible to encourage broad adoption by the Haskell community. It provides strong enough guarantees about compiled Haskell code for more advance secure systems to be built on top of Haskell, using techniques such as information flow control security or encrypted computations. These techniques combined with Safe Haskell make Haskell a great language for building reliable, secure multi-party systems today. Particularly relevant with the growing power of web applications and the platform nature of many web sites.

BODY

Safe Haskell enforces type safety in Haskell and adds a notion of trust to modules. This allows security to be built at the language level.

REFERENCES

[1] D. Terei, S. Marlow, S. Peyton Jones, and D. Mazières. Safe Haskell. In *Haskell Symposium.* ACM SIGPLAN, 2012.

Volume 1 of Tiny Transactions on Computer Science