LangSec SPW'15:
words from the organizers

Type theory connection

- Parsers: <u>Eliminating</u> strings, <u>introducing</u> types for the rest of the program

  - consuming strings, constructing well-typed objects

- Entire classes of weaknesses are due to incorrect string elimination (misrecognition) or creation of wrong types

  - E.g.: X.509 Bignum emitted into a narrow integer despite its correct recognition as a string (string is eliminated, fixnum is introduced instead of a bignum).

Verification connection

- Parsers enable proofs by creating well-formed <u>pre-conditions</u> for program correctness proofs

  - verified compilers: from ASTs onward

- Verified parsers are rare (exploitable parsers are not :))

  - Verifying parsers has the biggest security gain potential (i.e., almost every crafted input vuln)

More themes this year

- Correctness-security gap

- Heap allocator semantics

- Automatic synthesis of grammars

- Lowering parser verification costs

- Rust, Unparsers, Session languages & more!