LangSec 2027

Thirteenth IEEE S&P Workshop on Language-Theoretic Security

The 13th *IEEE* S&P Workshop on *Language-Theoretic Security (LangSec)* co-located with the 48th *IEEE Symposium on Security and Privacy* solicits contributions related to the growing field of language-theoretic security.

Venue: Le Centre Sheraton Montreal Hotel, Montreal, Canada.
Date: May 20, 2027
Website: `http://langsec.org/spw27`

About LangSec

The Language-theoretic approach (LangSec) is mission assurance for connected software and hardware exposed to attacks via malicious inputs. LangSec offers a practical data and code co-design methodology accessible to developers of everyday software.

In a nutshell, since any software that processes inputs is, in fact, an interpreter, and any inputs are thus its programs, input validation is not essentially different from program verification. To be trustworthy, input validation must therefore be grounded in models that describe precisely what the properties of valid inputs are. Luckily, such models exist, and can be made accessible to regular developers.

The language-theoretic approach (LangSec) is mission assurance for connected software and hardware exposed to attacks via malicious inputs–through a practical data and code co-design methodology and filtering of legacy formats down to safe subsets.

LangSec regards the Internet insecurity epidemic as a consequence of ad hoc input handling. LangSec posits that the only path to trustworthy computer software that takes untrusted inputs is treating all valid or expected inputs as a formal language, and the respective input-handling routine as a parser for that language. Only then can any correctness guarantees be assured for the input-handling code. Ambiguity of message/protocol specification is insecurity; ad hoc parsing is an engine of exploitation; overly complex syntax can make judging security properties of input impractical or even undecidable.

LangSec explains why ad hoc *input sanitization*, *sanity checking*, and other advice to be more careful with inputs is not enough, and why numerous secure programming initiatives have not ended input-driven exploitation. LangSec is also a code and protocol auditing methodology.

Treating input-handling code as an automaton allows the defender to reason about its behavior. The more limited computational power of the automaton, the easier the reasoning. The root cause of many bugs, memory corruptions, and exploitation is trying to validate inputs with inappropriate automata (e.g., much of XSS is due to "validating" context-free HTML with regexps). The recognizer automaton should be just as powerful as warranted by the message format, and no more; unnecessary complexity is computational power given to the attacker.

The 13th edition of the workshop will continue the tradition and further focus on methodologies (1) that can infer formal language specifications from samples of electronic data, (2) that can generate secure parsers from formal specifications of electronic data, and (3) that describe the complexity hierarchy of verifying parser implementations.

Important Dates (tentative)

- Paper submissions due: January 22, 2027, AOE
- Research Reports, Panels, and Proof-of-concept submissions due: February 17, 2027, AOE
- Notification to authors: March 7, 2027
- Final files due: April 7, 2027

Full papers and presentations from previous workshops freely available at `http://langsec.org`.