**LangSec: A Workshop on Language Theoretic Security**
**Date & Venue:** Thursday May 21, 2015, Fairmont Hotel, San Jose, CA
**URL:** `http://spw15.langsec.org/` (1st LangSec workshop: `http://spw14.langsec.org/`)

# Call for Papers

*Language-theoretic security* (LangSec) is a software design and programming approach that focuses on formally correct and verifiable input handling throughout all phases of the software development lifecycle. In doing so, it offers a practical method of *assurance* of software free from broad and currently dominant classes of bugs and vulnerabilities related to incorrect parsing and interpretation of messages between software components (packets, protocol messages, file formats, function parameters, etc.)

LangSec aims to (1) produce verifiable recognizers, free of typical classes of ad-hoc parsing bugs, (2) produce verifiable, composable implementations of distributed systems that ensure equivalent parsing of messages by all components and eliminate exploitable differences in message interpretation by the elements of a distributed system, and (3) mitigate the common risks of ungoverned development by explicitly exposing the processing dependencies on the parsed input.

As a design philosophy, LangSec focuses on a particular choice of verification trade-offs: namely, *correctness and computational equivalence of input processors*. It is informed by the collective experience of the exploit development community, since exploitation is practical exploration of the space of unanticipated state, inasmuch as defense is about its prevention or containment.

LangSec offers a unifying explanation for the existence of vulnerabilities and their continual perpetuation under current software design practices despite massive efforts at defining secure development practices. In short, the existence of exploitable bugs is a consequence of software designs that make verification and comprehensive testing infeasible and *undecidable in the formal sense.* LangSec thus sheds light on the continued existence of software flaws by highlighting the connection between fundamental Computer Science concepts (language theory, computability) and faulty software designs.

Bugs in input processing (wherever input is taken at a software module's communication boundary) clearly dominate other kinds of bugs. Hence the first order of business in securing software that does any communication is ensuring that no unanticipated state is entered, and no unexpected computation occurs while consuming inputs. In practice, however, such code is often ad-hoc and lacks a clear, formal language-theoretic definition of valid payloads. What's worse, inputs are "checked" with recognizers that cannot possibly accept or reject them correctly, e.g., context-free formats with regular expressions. In such cases, subsequent code assumes properties that couldn't possibly have been checked, and thus cannot be trusted to abide by their specification. Non-existence of unexpected computation is then highly unlikely, and unanticipated state conditions proliferate.

## Important Dates (tentative)

**Submissions due:** 15 January 2015, 11:59 PM Pacific
**Research Reports, Panels, and Proof-of-concept submissions due:** 30 January 2015, 11:59 PM Pacific
**Notification to authors::** 15 February 2015
**Final files due:** 5 March 2015

The LangSec workshop solicits contributions of research papers, panel proposals, proof-of-concept software, and industry case studies related to the growing area of language–theoretic security.

Submissions should be in the PDF file format and made via EasyChair. Submissions must not be anonymized. The confidentiality of submissions will be protected as is customary, but submissions with non-disclosure agreements or forms attached will be returned without review.

## Research Papers

The LangSec PC encourages submission of research papers from academia, industry, and government. There is no hard maximum page limit, but length should be justified by the content and quality of the text. The PC expects research papers to vary between 4 and 15 pages in length. Shorter papers are encouraged, but longer papers that document high-quality or extensive experimentation are very much in scope. Research papers are encouraged to address some of the topics listed below, but the list is not exhaustive:

### LangSec design:

1. *LangSec Stack*: systems architectures and designs based on LangSec principles
2. computer languages, file formats, and network protocols built on LangSec principles
3. science of protocol design: layering, fragmentation and re-assembly, extensibility, etc.
4. re-engineering efforts of existing languages, formats, and protocols to reduce computational power
5. novel system and API designs for isolation and separation of parsers and processing
6. architectural constructs for enforcing limits on computational complexity

### Offensive research:

1. exploit programming as an engineering discipline
2. structured techniques for building weird machines
3. identification of LangSec anti-patterns; certification of absence
4. formalization of vulnerabilities and exploits in terms of language theory

### Software construction & verification:

1. type safety; efficient runtime type checking
2. small languages
3. parser generators
4. embedding runtime language recognizers
5. methods and techniques for practical assurance
6. parser proof-of-equivalence in distributed systems

### LangSec engineering & practice:

1. LangSec case studies of successes and failures
2. how to evaluate software systems or packages for their adherence to LangSec principles
3. LangSec measurement studies of systems or data sets
4. comprehensive taxonomies of LangSec phenomena
5. empirical data on programming language features/programming styles that effect bug introduction rates (e.g., syntactic redundancy)

The PC expects that topics should cover recent LangSec–related advances or make the connection between research and practical assurance through computability theory. The LangSec PC particularly encourages studies that deal with controlling LangSec anti-patterns:

1. Ad-hoc notions of input validity.
2. Parser differentials: mutual misinterpretation between system components.
3. Mixing of input recognition and processing (a.k.a. “Shotgun parsers”).
4. Ungoverned development: adding new features / language specification drift.

### Research Reports

The LangSec PC encourages submissions of condensed research reports, especially from security practitioners. Submissions must describe LangSec-related research projects and their preliminary results.

### Proof-of-Concept

The LangSec PC encourages submissions that discuss actual implementations, prototypes, and proofs-of-concept. The resulting talk must not be a product pitch or a product manual, but the PC expects that the demonstration should enlighten and educate the audience to the extent that the audience could subsequently apply the tool or system in their own research or work. Proof-of-concept submissions are encouraged to include in their paper submission links to videos or other media demonstrating the project.

### Panels

The LangSec PC seeks submissions for interesting and lively panel topics. Panel submissions should identify the panel moderator and two to three panelists that have agreed to participate, along with short position statements on the panel's core question or assertion. Panel proposals should be at most four pages in length. Panels will be scribed, and the notes published in the proceedings alongside the panelist statements.

### Program Committee:

Sergey Bratus (Dartmouth College)
Jon Callas (Silent Circle)
Thomas Dullien (Google)
Alex Gantman (Qualcomm)
Dan Geer (In-Q-Tel)
Robert Graham (Errata Security)
David Grawrock (Intel)
Peter Gutmann (University of Auckland)
Felix Lindner (Recurity Labs / Phenoelit)
Michael E. Locasto (University of Calgary)
Greg Morrisett (Harvard University)
Collin Mulliner (Northeastern University)
Meredith L. Patterson (Nuance Communications / Upstanding Hackers, Inc.)
Sean W. Smith (Dartmouth College)
Julien Vanegue (Bloomberg)
Jesse Walker (Intel)
Samuel Weber (Software Engineering Institute, CMU)
Stefano Zanero (Politecnico di Milano University)

### Organizing Committee:

Sergey Bratus (Dartmouth College)
Daniel 'TQ' Hirsch (P3KI GmbH.)
Felix 'FX' Lindner (Recurity Labs / Phenoelit)
Michael E. Locasto (University of Calgary)
Meredith L. Patterson (Nuance Communications / Upstanding Hackers, Inc.)
Anna Shubina (Dartmouth College)