Compositional Security

PIs: Anupam Datta, Limin Jia and Jeannette Wing

Carnegie Mellon University

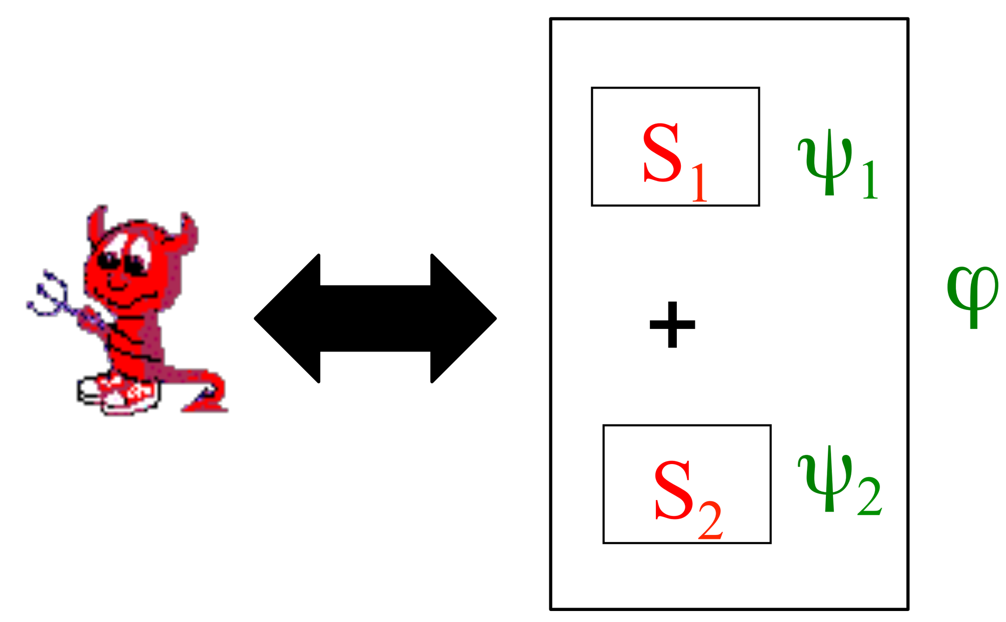
http://www.andrew.cmu.edu/user/danupam/compositional-security.html

The goal of this project is to develop a general theory of compositional security that can support the construction and analysis of secure systems

- Identify composition operators for systems, adversaries and properties.
- Develop compositional reasoning principles
- Apply theory to improve Web and hypervisor security

- Prior work
  - Protocol composition [Datta, Derek, Durgin, Mitchell, Pavlovic, Roy, …]
  - General first-order software system composition in the presence of *interface-confined adversaries* [Garg, Franklin, Kaynar, Datta]
- Currently: Higher-order functions (code is first-class data) [Jia, Garg, Datta]

Do S_1+ S_2 satisfy a global security property φ based on local properties ψ_1 of S_1 and ψ_2 of S_2 that are checkable separately?



Approach

- System abstraction: Model the system using a programming language
  - Types specify the trace properties
  - Typing rules reason about compositions
  - Adversaries are confined to the set of interfaces (first-order)

- Reasoning principles
  - Local reasoning: $\vdash$ P : $\{\varphi\}$ (in the presence of adversaries)
  - Adversary: $\vdash$ A : $\{\varphi_A\}$ (given fixed set of interfaces)
  - Compositional reasoning:

$$\frac{\Gamma_1 \vdash \mathcal{P}_1 : \{\varphi_1\} \;\ldots\; \Gamma_n \vdash \mathcal{P}_n : \{\varphi_n\} \quad \vdash \mathcal{A} : \{\varphi_A\} \quad \vdash \Gamma_1 \;\ldots\; \vdash \Gamma_n}{\vdash \mathcal{P}_1 \mid \ldots \mid \mathcal{P}_n \mid \mathcal{A} : \{\varphi_1 \wedge \cdots \varphi_n \wedge \varphi_A\}}$$

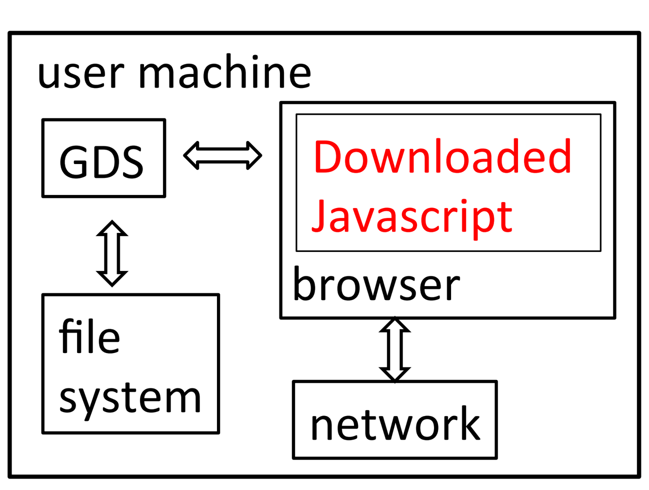
- Complex, increasingly mobile, software architecture requires reasoning about higher-order functions

Attackers can supply code using higher-order interfaces
- Interfaces that take code as input (callbacks)
- Interfaces that return code (script in webpages)

Types specify trace properties of interfaces
- $\{y{:}\,\tau\}\langle u_1, u_2, i\rangle\varphi$
  Computation returns a value of type τ, and if the computation executes between time u_1 and u_2 by thread I, the trace satisfies φ
- $(\tau_1 \to \tau_2) \to \tau_3$
- $\tau_1 \to (\tau_2 \to \tau_3)$

- **Case studies** [Datta, Garg, Jia, Sen, Wing]

Web security
- Reason about properties of (malicious) downloaded code (M), given the specifications of the interfaces that M is confined to
- The type assigned to M allows reasoning about systems that pass M around as data, and invoke it later



Hypervisor security: (guest OS and hyper-apps require higher-order reasoning principles)
- Core:
  initialization function, interrupt handling, memory virtualization
- Guest OS: (potentially malicious)
  confined to Hypervisor provided interfaces above
- Hyper-apps: (may not be trusted)
  register interrupt handlers
  confined to a set of interfaces core provides (different from guest)

2012 Science of Security Community Meeting
Nov. 29-30, 2012
National Harbor, MD
http://cps-vo.org/group/sosmtg

Carnegie Mellon University