

Temporal Aspects

<http://www.teasp.org>

G. Bruns, A. Jeffrey, R. Jagadeesan, J. Riely, C. Pitcher

Aspects For Security

Aspects have emerged as a powerful tool to accommodate change in the design and development of systems. Our research explores updates to the functionality of systems in a way that is *both* dynamic and safe, so as to not compromise the trustworthiness of the system. We focus on aspects used in security applications such as type enforcement and security automata.

Research Issues

AspectJ is not type-safe. The design for generics is in progress.

New Approach [SCP 2006]

Safety via source-level typing for parametric polymorphism.

Research Issues

Expressive aspect module languages are in development. The languages can express the temporal and flow constraints needed for access control and auditing in security.

But what are the modularity guarantees?

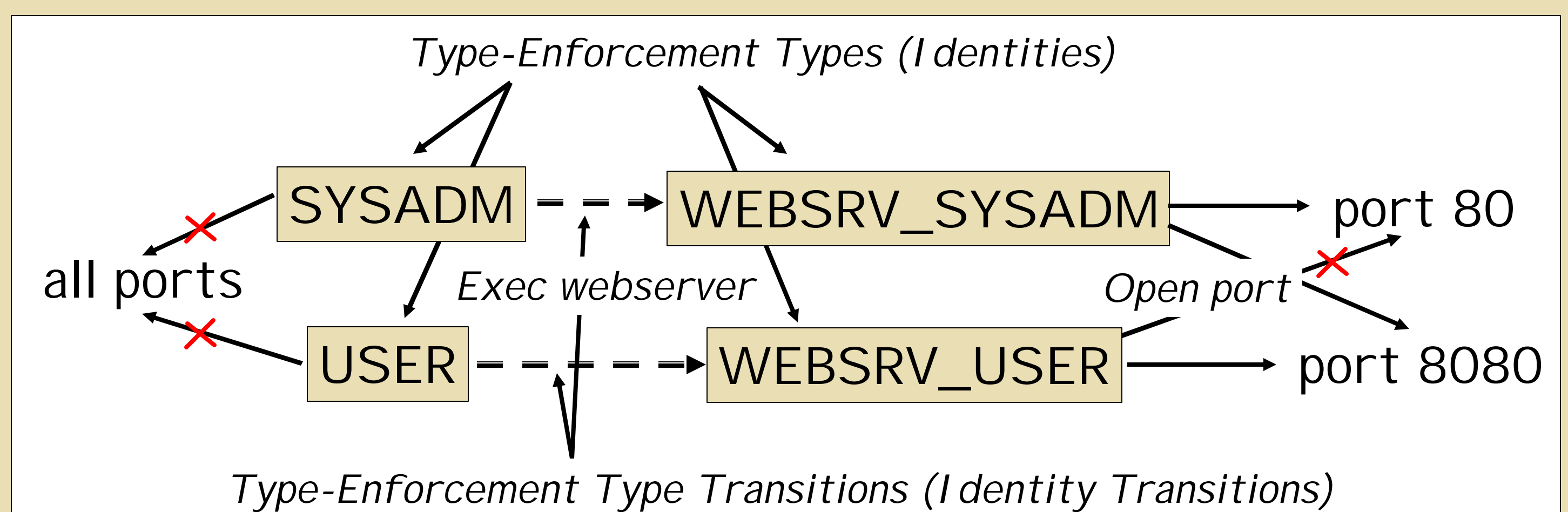
New Approach [AOSD 2007]

- Compositional reasoning to establish modularity principles.
- General reasoning methods to adapt imperative techniques to aspect-based languages.
- Examples demonstrate that aspects enforce desired security guarantees.

Example Scenario A type-enforcement policy allows a sysadmin to run a webserver on ports 80 and 8080, but other users are restricted to port 8080. Sysadmin and other users can only open ports by running the webserver---direct access is prohibited---causing a

type-enforcement transition. Policy enforced via aspects.

Property Code run by user cannot send/receive on port 80: formalized as contextual equivalence of two programs with aspects, and proven by *open* bisimulation.



Key Technical Points

- Contexts are powerful in dynamic-aspect languages, e.g., dynamic advice subsumes state.
- Ability to hide and create new pointcuts provides flexible control and leads to a rich theory.
- LTS for aspect programs permits addition of advice for public pointcuts by the environment.
- *Open* bisimilarity is sound and complete w.r.t. contextual equivalence, facilitating compositional reasoning.