The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Department of Homeland Security, Air Force Research Laboratory or the U.S. Government.
Team Profile

HRL Laboratories, LLC
George Kuan (PI)  Aleksey Nogin

- Formerly Hughes Research Laboratories (est. 1948)
- Formed as a Limited Liability Company (LLC), 1997
- R&D for The Boeing Company and General Motors
- Government and commercial contracts
- AS9100 accredited / DoD Trusted Foundry
- 250,000 square feet of lab space
- 10,000-square-foot Class 10 clean room
- Located on 72 acres in Malibu, CA

Dave Naumann (PI)
Andrey Chudnov

- Established 1870
- Located in Hoboken, NJ
- Also online and in DC
- Schools of: Engineering and Science; Technology Management; Systems and Enterprises
- Designated a National Center of Academic Excellence in Information Assurance Education (CAE) and Research (CAE-R)
- DoD National Center of Excellence in Systems Engineering Research
- DHS National Center of Excellence in Port Security
- Ranked #3 among US research universities for high ROI on research investment (Forbes.com)
SDV/SLAM verified one-size-fits-all safety properties. Security is different: one party’s feature may be another party’s vulnerability. Developers may favor features and performance over security.

Another problem: The rest of us can’t easily get 3rd-party developers to run verification tools with our security properties of interest.
We need to give the software acquirer the ability to verify custom security specifications.

**Approach (1) - AVAP**

- **Supplier**
  - Supplier Security Policy Specifications
  - Source Code

- **Acquirer**
  - Acquirer Security Policy Specifications

- **HRL Automated Security Policy Prover**
- **Vulnerability Feedback**
- **Proof-Carrying Code Framework**
- **Symbolic Execution Based Testing and Inlined Runtime Monitor**

- **Acquirer-Side Spec Verification**
- **Secure Software Binaries**
- **HRL Automated Security Check Transformation**
- **Software with attached mathematical proofs of supplier security**

© 2013, HRL Laboratories, LLC. All rights reserved.
Proof-Carrying Code (Necula and Lee ’97) takes advantage of the observation that verifying a proof is easier and faster than generating one.

Source: Appel et al ‘03
Approach (3)

Then, the **Acquirer**-side tool uses the proof of supplier’s spec to help prove acquirer’s spec.

Acquirer-side verification tailors verification to relevant security properties instead of requiring a one-size-fits-all security specification.

The Acquirer no longer depends on the Supplier (3rd party developers) to agree on and verify the same specification.

From the HRL compiler pass

The **Acquirer**-side tool first verifies proof in the context of the code.

Supplier’s spec

\[ R_s \rightarrow E_s \]

Supplier’s spec proof

Acquirer’s spec

\[ R_a \rightarrow E_a \]

The **Acquirer** no longer depends on the Supplier (3rd party developers) to agree on and verify the same specification.
(X,Y,Z)
Input space is divided into a set of equivalence classes, each defined by the region of input space that travels the same path through execution state space.
Benefits

• For the Software Assurance community, Tunable Info Flow enables verification of Acquirer specs by the Acquirer without having to divulge the specs to the Supplier or a 3rd-party

• It empowers the Acquirer to check the most relevant information flow security spec and simultaneously simplify verification by taking into account the Acquirer’s implicit assumptions

• Highly expressive framework for encoding properties

• Can help enable information flow-preserving compilation

• Portable across virtual machines and just-in-time compilers

• Takes advantage of existing compiler optimizations
Current Status

- We have designed and implemented prototypes for the Compiler Pass and Proof Checking Tools.
- We have also designed the Runtime Monitoring Tool and adapted a symbolic executor to propagate information flow security tags.
- Designed an information flow specification contract language with novel features motivated by our analysis of vulnerabilities
- Theory and implementation technique for checking specification contract refinement
Next Steps

- Runtime Monitoring for Information Flow
- Larger-scale performance analysis
- Automated feedback mechanisms
- Transition activities
Contact Information

For more information:
George Kuan
gkuan@hrl.com