Manifest Security

NSF-0715936 (CMU) & NSF-0716469 (Penn)


Karl Crary (CMU)


Robert Harper (CMU)


Frank Pfenning (CMU)


Benjamin C. Pierce (Penn)


Stephanie Weirich (Penn)

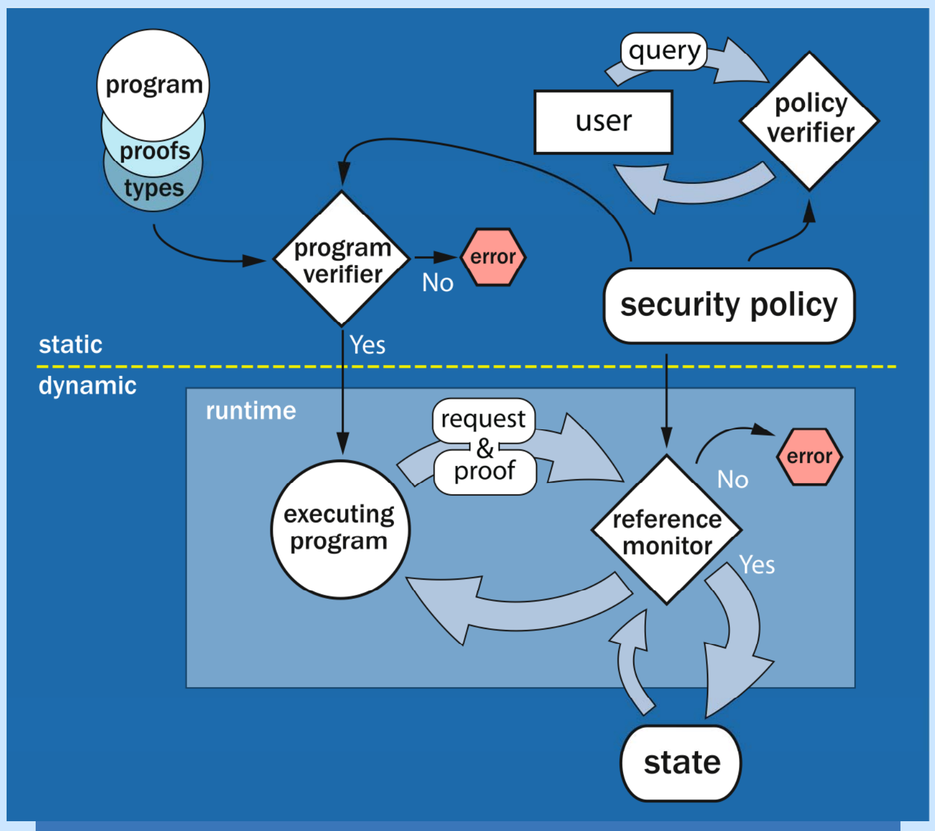

Steve Zdancewic (Penn)

Security for extensible software platforms:

Manifest security is a new architectural principle where the security properties of a system are made fully explicit and checkable.

Security policies are explicitly represented within a security logic and compliance with the policy is manifest in proof objects. Proofs can be externally verified to provide an audit trail, and the policies are cleanly separated from the enforcement mechanisms. The proof-checking TCB can be made very small.

Programming support for development of applications with manifest security properties.



Manifest Security Architecture

Approach and Impact

Novelties of the approach

- New formal security logic
- Proof-carrying run-time system
- Machine-checked metatheory

Expected Research Impact

- Practical policy specification
- Language implementation
- Web browser case study

$$\frac{}{\Gamma, P\ true \Longrightarrow P\ true} \qquad \frac{\Gamma \Longrightarrow A\ true}{\Gamma \Longrightarrow K\ affirms\ A}$$

$$\frac{\Gamma, A\ true \Longrightarrow K\ affirms\ C}{\Gamma, K\ affirms\ A \Longrightarrow K\ affirms\ C} \qquad \frac{\Gamma \Longrightarrow K\ affirms\ A}{\Gamma \Longrightarrow \langle K\rangle A\ true}$$

$$\frac{\Gamma, A\ true \Longrightarrow K\ affirms\ C}{\Gamma, \langle K\rangle A\ true \Longrightarrow K\ affirms\ C} \qquad \frac{\Gamma, A\ true \Longrightarrow J}{\Gamma, K\ knows\ A \Longrightarrow J}$$

$$\frac{\Gamma|_K \Longrightarrow A\ true}{\Gamma \Longrightarrow K\ knows\ A} \qquad \frac{\Gamma \Longrightarrow K\ knows\ A}{\Gamma \Longrightarrow [\![K]\!]A\ true} \qquad \frac{\Gamma, K\ knows\ A \Longrightarrow J}{\Gamma, [\![K]\!]A\ true \Longrightarrow J}$$

A (fragment of a) logic of affirmation and knowledge. These operators allow the specification of rich authorization and Information-flow policies. Affirmation is used to allow principals to make policy assertions; knowledge is used to restrict how secrets may be manipulated by the system.

U = the user principal
I = Identifier of the user at site S
P = a password
P' = a password hint
S = site to which the password applies
F = persistent storage (file system)

• If U affirms that its ID and password for site S are I and P, then the file system may store that data (in an encrypted fashion):
<U>id_and_pwd(S,I,P,P') ⊃ [F]<U>id_and_pwd(S,I,P,P')

• If the file system has stored U's password for site S, then S may be sent this information:
[F]<U>id_and_pwd(S,I,P,P') ⊃ [S]password(I,P)

• U can see its ID, password length, and hint, but not the password itself:
[F]<U>id_and_pwd(H,I,P,P') & length(P,K) ⊃ [U]id_and_hint(H,N,K,P')

A simple password manager plugin policy example. The policy restricts which web sites may be sent a user's password information. Using additional policy features, such as linearity, the policy can be refined to specify under what conditions the user may/must change passwords.

National Science Foundation
WHERE DISCOVERIES BEGIN

NSF Cyber Trust Principal Investigators Meeting
March 16-18, 2008
New Haven, CT

Carnegie Mellon Penn University of Pennsylvania