

PROJECT PROFILE

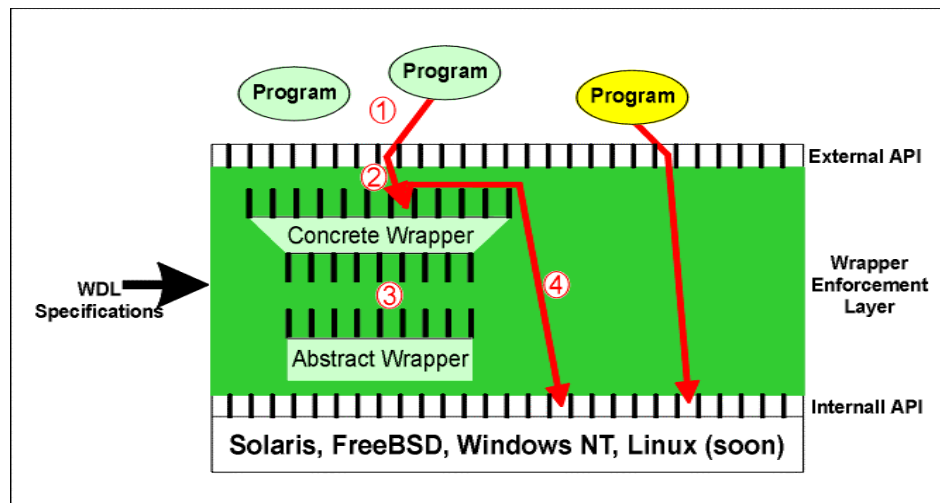
Generic Software Wrappers for Security and Reliability

Problem

Today's large-scale information systems increasingly combine an array of independently developed Commercial Off-The-Shelf (COTS) software components such as programs, linkable code libraries, and network applets (e.g., CORBA or Java). Unfortunately, however, although conventional software composition mechanisms (e.g., network protocols, dynamic libraries, system APIs) supply the required "glue" for combining such components into large systems, they provide very weak inter-component boundaries. Consequently, an entire critical system may be vulnerable to failures or security compromises within a single component. One solution might be to base critical systems only on high-assurance trusted components, but such components rarely are available. Network Associates' NAI Labs is investigating a very promising and practical potential security solution for large-scale systems comprising numerous individual COTS components.

Solution

Under DARPA funding, NAI Labs is developing software "wrapping" technology to significantly increase the security and reliability of large software systems composed of standardized software components. These generic software wrappers intercept and augment component interactions to implement practical security (e.g., restricting, filtering) and reliability (e.g., redundancy, crash data recovery) policies.



Details

The figure below illustrates the architecture of a wrapper-enforcing system. Programs running "unwrapped" perform as usual; any system requests go directly to the operating environment. However, selected system requests from a wrapped program (step 1) are intercepted by a wrapper that understands the system API (step 2). This *concrete wrapper* may in turn generate events intercepted by a more *abstract wrapper* (step 3). If the event has not been denied, it passes down to the system's internal API (step 4).

Research Focus

Two Fundamental Challenges

Our research is focusing on two fundamental challenges for practically deploying non-bypassable wrappers:

- How to cost-effectively specify security policies as event interceptions; and
- How to support wrappers using COTS operating systems and network execution environments (e.g., UNIX, Windows NT)

To specify security policies as event interceptions, our research is formulating a Wrapper Definition Language (WDL) to specify lightweight, portable software wrappers that can be used to provide security and reliability to generic software components. The goal of WDL is to make the specification of wrappers as easy and concise as possible.

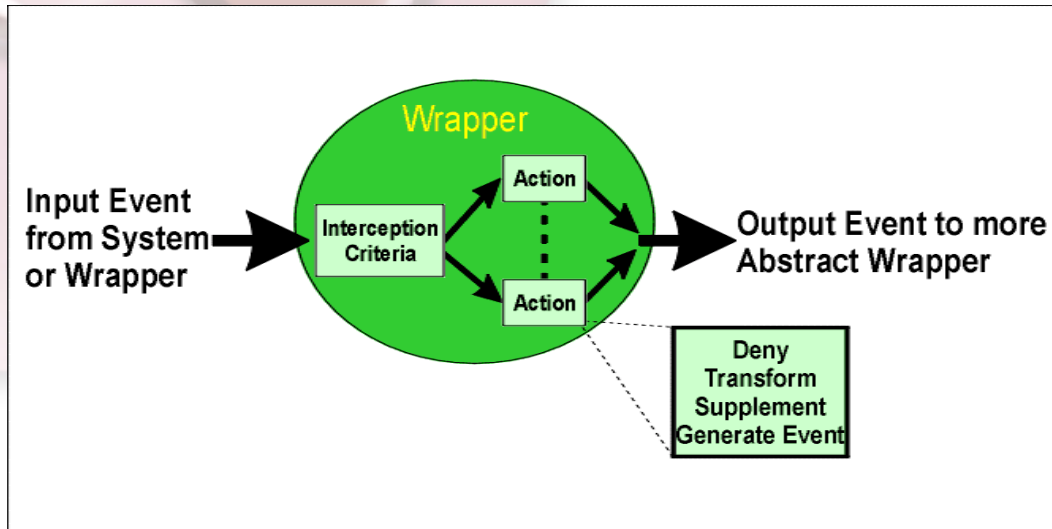
To support wrappers, our research has developed a Wrapper Support Interface (WSI) and a Wrapper Support Subsystem (WSS). The WSI specifies all operating system services required by wrappers; the WSS implements the WSI. The WSI and WSS have been developed for inclusion in both mainstream UNIX systems (currently FreeBSD 2.2.x and 3.x and Sun Solaris 2.6, with development in progress for Linux) and the Windows NT 4 runtime environment.

- Lee Badger
Chief Scientist,
NAI Labs

Details (continued)

A central feature of our approach is that highly abstract wrappers may directly express security policies of interest (e.g., Biba, Clark/Wilson) and that more concrete wrappers may translate a particular system's API or use by abstract wrappers. By placing wrapper logic primarily in abstract wrappers, our research seeks to make wrappers relatively reusable and portable between execution environments. Furthermore, by showing WDL wrappers that run in the UNIX and Windows prototype systems, our research seeks to demonstrate that WDL wrappers are not specific to a single system or architecture, but are suitable for increasing the security and reliability of large-scale heterogeneous software systems in general.

The figure below graphically depicts a wrapper written in WDL and traces its key behaviors. The wrapper specifies *interception criteria* that determine the events to be intercepted by the wrapper. For each event intercepted, an *action* is performed; for example, the event is denied, parameters are transformed, the event's functionality is supplemented (e.g., data is encrypted), or a new event is generated for a possible consumption by another wrapper.



Additionally, we have been focusing research on using wrappers to implement intrusion detection techniques. Wrappers provide better access to system data, including all system call parameters, for intrusion detection and the ability to respond faster, stopping attacks at the first system call at which they are detected. We have written wrappers that successfully implement specification-based and sequence-based intrusion detection concisely and with low overhead.

Additional Information

For additional technical information regarding Generic Software Wrappers, contact Lee Badger at 443-259-2382 (lbadger@nai.com); Mark Feldman at 443-259-2347 (mfeldman@nai.com), or download reports, papers, other documentation, and the current Generic Software WrappersToolkit from: <ftp://ftp.tislabs.com/pub/wrappers>.

1/5/01



Who's watching your network

For more information on NAI Labs, contact your authorized NAI Sales Representative, or visit <http://www.nailabs.com>.

CORPORATE Headquarters

3965 Freedom Circle
Santa Clara, CA 95054-1203
Tel (800) 764-3337*
Fax (888) 203-9258

*Call for additional Worldwide Sales Offices