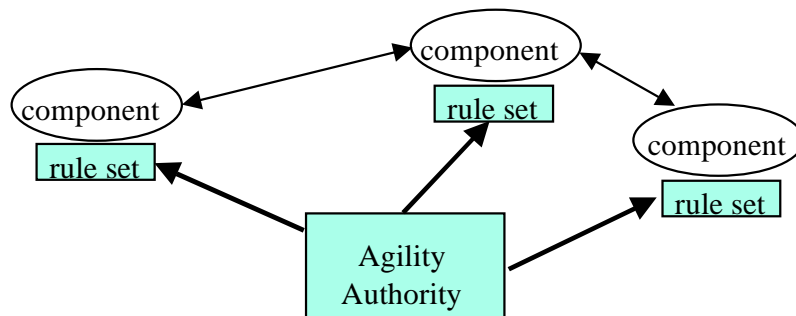


PROJECT PROFILE

Security Agility for Dynamic Execution Environments

In rapidly changing environments, software components must be able to respond flexibly and quickly to changing resource availabilities and security policies. Network Associates, Inc. Laboratories (NAI Labs) research using Domain and Type Enforcement (DTE), a flexible form of centralized access control, has highlighted an important characteristic of current-generation software: unexpected security policy changes can have catastrophic side-effects, often causing software components to freeze, crash, fail to support security rules, and otherwise malfunction. For mission critical operations, such as command and control systems, this behavior is unacceptable: such systems require software components that gracefully reconfigure to changing security requirements with minimal impact on service availability.



Under DARPA funding, this research is developing a software flexibility technique encapsulated in a toolkit that extends the functionality of software components to accommodate the dynamic security properties of their environment. An agile software component (process) is aware of its security environment, is able to enforce "its part" of a more global policy, and contains internal mechanisms that can adapt its functionality to reliably conform to authorized policy changes. The above figure abstractly shows our strategy: security rule sets are added to software components. These rule sets provide components with built-in knowledge of security policies, models, and mechanisms and the means to adapt to policy changes. In addition to their mission-specific behaviors, agile software components interact with a policy update manager, the Agility Authority. The Agility Authority distributes authorized security policy change requests to agile software components, which then evaluate their current behavior with respect to the new security policy requirements and dynamically reconfigure. In response to a policy change, a security agile software component may take a number of actions, for example, suspending connections, revoking access, reestablishing connections, changing cryptographic algorithms, reacquiring resources, or accepting additional security enforcement responsibilities.

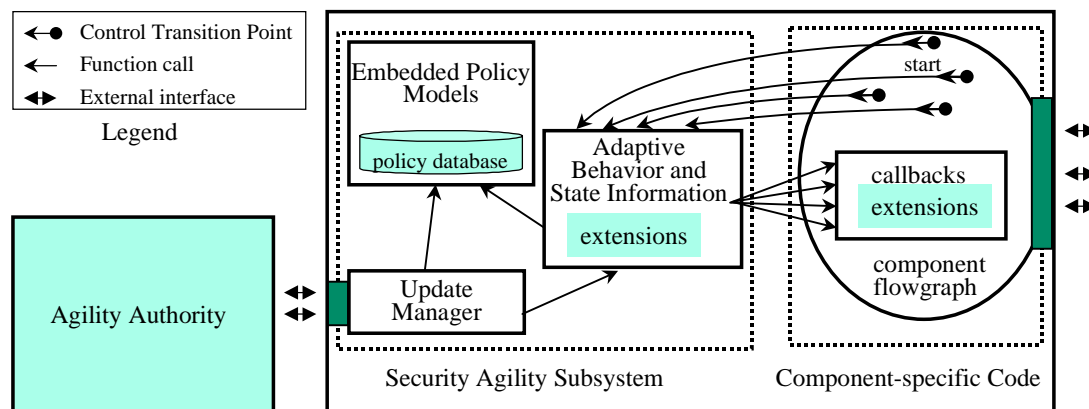
Research Focus

Agility Authority

The Agility Authority distributes authorized security policy change requests to agile software components, which then evaluate their current behavior with respect to the new security policy requirements and dynamically reconfigure. In particular, security agile software components carefully track their use of resources, maintain security attributes for resources, and validate that current resource usage is consistent with new security rules. In response to a policy change, a security agile software component may take a number of actions, for example, suspending connections, revoking access, re-establishing connections, changing cryptographic algorithms, reacquiring resources, or accepting additional security enforcement responsibilities.

- Mike Petkac
Principal Investigator
Secure Execution
Environments Group

The figure that follows shows our architecture for a security agile component. The component-specific code primarily consists of the component's mission-specific algorithm. As the algorithm executes, control is transferred to the security agility subsystem at critical junctures called Control Transition Points (CTPs). The CTPs provide the security agility subsystem with a trace of the component's behavior. The subsystem uses this trace to associate security attributes with component-specific resources, to perform security mediation, and to trigger appropriate security services (e.g., cryptography) on behalf of the component. In order to perform component-specific security recovery, the component-specific code is extended with callback functions. The callback functions are invoked by the security agility subsystem to accomplish component-specific dynamic security reconfiguration for tasks such as parsing component-specific file formats and traversing component-specific data structures.



A prototype toolkit was constructed and integrated with various UNIX-based system components to demonstrate these techniques. Though the toolkit leveraged our DTE technology to provide validated security policy tools, including dynamic policy reconfiguration, toolkit extensibility for additional security policy specifications and portability to other operating systems have been primary research goals. To assist in this effort, an application-level policy model has been developed to explore security agility in the FreeBSD, Linux, and Windows NT environments. An initial toolkit release for the FreeBSD 3.2 operating system (<ftp://ftp.tislabs.com/pub/agility>) was made available in September 1999. In September 2000, an updated toolkit release for the Linux environment will be available that is simpler to integrate and configure than its predecessor. Based on the updated toolkit, we have also proposed a method in which security agility can be used to help automate flexible host-based response to intrusions that we will present in December at the 16th Annual Computer Security Applications Conference in New Orleans, Louisiana.

Additional Information

For additional technical information on NAI Labs security agility research, contact Mark Feldman (mfeldman@nai.com) at 443-259-2347 or visit our Web page at: <http://www.pgp.com/research/nailabs/secure-execution/agility.asp>.

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