Abstract
Security wrappers for Windows NT/2000 encapsulate program execution to monitor their run-time behavior and ensure that they don't do anything harmful. Wrappers have been developed to ensure that email attachments can be safely opened, that web content can be safely viewed, downloaded, and executed, and that active content in office documents can be safely executed.

The wrappers can be directed to allow the execution of operations that violate security rules to safely continue by virtualizing their operation so that the effects are contained within the executing process and are not visible to any other process.

This ensures that processes run within their prescribed security policy (relative to the real resources protected by that security policy) without user intervention or control. Violations of that security policy are automatically and transparently contained without user involvement or knowledge.

This fully autonomic safety policy eliminates the configuration problem that has plagued existing behavior monitoring systems.

Demo Description
We will demonstrate security wrappers for Windows NT/2000 that encapsulate program execution to monitor their run-time behavior and ensure that they don't do anything harmful. Wrappers have been developed to ensure that email attachments can be safely opened, that web content can be safely viewed, downloaded, and executed, and that active content in office documents can be safely executed.

These wrappers detect violations of process-specific rules establishing the acceptable (and safe) behavior of the wrapped processes relative to six resources: the file system, the system registry, inter-host communication, process spawning, devices, and establishing COM connections.

The wrappers can determine whether an operation is being performed by the native application or by active content within the email attachment or downloaded document (including Active-X components or arbitrary executables) and applies a different (and presumably more stringent) set of rules to the latter. It enables active content to be used throughout MS Office, Outlook, and Internet Explorer without compromising security.

When attempted violations are detected, the security policy determines whether the violation is automatically handled or the user is notified and informed of the severity of the violation. In the latter case, the user determines whether to allow or prohibit the offending operation. In both cases the violation, the system's or user's response, and the initiating email and attachment - obtained from the email client – are logged.

Violations can be automatically denied by the system to avoid the need for user involvement in the handling of the violation. But while such denial of service ensures the safety of the execution, it affects the behavior of the application and often causes it to break and/or terminate.

The wrappers introduce another option, called Contained Execution, that allows the execution of operations that violate security rules to safely continue by virtualizing their operation so that the effects are contained within the executing process and are not visible to any other process.

Its use ensures that processes run within their prescribed security policy (relative to the real resources protected by that security policy) without user intervention or control. Violations of that security policy are automatically and transparently contained without user involvement or knowledge.

This Contained Execution capability can be used to create a desktop user environment that is inherently safe for executing imported content of all kinds by limiting and managing the propagation of effects from these potentially malicious processes to the rest of the system. Several important classes of applications (such as viewers, importers, and editors) can be run under a single safety policy that contains all effects except for explicitly authorized information extraction (via the built-in copy-and-paste or SaveAs operations). This fully autonomic safety policy eliminates the configuration problem that has plagued existing behavior monitoring systems.