# Veracode Input for Developing a Framework to Improve Critical Infrastructure Cybersecurity

## Background

Veracode provides automated and manual application security testing for over 300 organizations in the US and abroad. Our customers range in size from small independent software vendors to large industrial corporations that are part of the military supply chain. Our US Government customers include: Federal Aviation Administration, US Army, and In-Q-Tel. We provide static and dynamic application security testing for organization that need to secure the applications they are building and the applications they are procuring.

Veracode has been regularly invited to share our application security expertise at government events such as DHS/DoD/NIST Software Assurance Forums. We have participated in NIST SAMATE for 3 years. Veracode is an active contributor to DHS sponsored standards: Common Weakness Enumeration (CWE), the Common Weakness Scoring System (CWSS), and the CWE/SANS Top 25 Programming Errors.

Veracode is a strong supporter of objective and independent measures of application security. Community developed lists of the most serious programming errors that lead to application vulnerabilities, such as the OWASP Top 10 and the CWE/SANS Top 25 create a minimum due care standard that all software should be held against. Independent testing assures the operators of software that the developers of the software were held accountable to these standards.

The advancement in the accuracy and vulnerability class coverage of automated testing technology, both static analysis and dynamic analysis, has set a new minimum bar for application security. No application should be deployed and operated without proof that testing was independently performed and the top programming errors remediated. Many of Veracode's customers, including some of the largest software and industrial companies in the world have established application risk management programs that use automated application security testing to assure the security of the software they are developing and procuring.

## Suggestions

Software that is purchased or built by critical infrastructure operators should have a reasonable protective measures applied during the software development process. An example of these protective measures is automated and manual security testing of the software before it is placed into production operation. Evidence that independant testing was performed and defects that are specified in well defined public standards such as the CWE/SANS Top 25 [ref: http://cwe.mitre.org/top25/] list were remediated. This evidence should be delivered with the software. Use the NIST definitions for the assurance of systems to require different levels of testing.

**SUGGESTION #1:**

**Have well defined requirements of *independent application security testing against the weaknesses defined by the current CWE/SANS Top 25*. Use a reference to the CWE/SANS Top 25 as it is updated annually to maintain currency with relevant risks. Automated testing is available at reasonable cost so it should be *a requirement for applications that are components of MEDIUM assurance systems in addition to applications that are components of HIGH assurance systems. Applications that are components of HIGH assurance systems should require further manual testing.* Independence of test execution, including static analysis, is critical because if members of the development team create the test results they are free to dismiss real issues as false positives or not applicable. This leads to vulnerabilities in the software being ignored.**

There is a need for the scope of controls no not just be focused on the implementation of security controls themselves but for the implementation of all software. It is clear that flaws in business software are being used to bypass security controls so even if the security controls are implemented flawlessly the flaws in the business software can be used by attackers. This is the trend we are seeing with SQL Injection attacks in web applications being used in the Night Dragon attacks on energy companies [ref: http://www.mcafee.com/us/resources/white-papers/wp-global-energy-cyberattacks-night-dragon.pdf] to bypass perimeter controls and flaws in desktop media viewing software being used for the same purpose in the RSA attacks [ref: http://blogs.rsa.com/anatomy-of-an-attack/]. All software, no matter its source (internally developed or purchased), needs to be subjected to a trustworthiness control based on its assurance level.

**SUGGESTION #2:**

**Controls should encompass all applications, not just those that are components of security controls. In addition it should encompass applications that are components of MEDIUM assurance systems with automated application security testing, in addition to applications that are components of HIGH assurance systems. Applications that are components of HIGH assurance systems should require further manual testing.**

## Contact Information

If there are any questions or need for clarification please contact:

Chris Wysopal

cwysopal@veracode.com

339-674-2703