4. Initial minimum requirements for testing software source code

Submitted by:
Jeff Williams, Veteran Application Security Industry Leader
jeff.williams@contrastsecurity.com
https://www.linkedin.com/in/planetlevel/
410-707-1487

We write to encourage NIST to promote an effective application security testing (AST) strategy that works at the speed and scale of modern software development.

Each of the authors of this position paper have over 20 years of experience with AST on critical applications including defense, elections, finance, utilities, airlines, and more. These efforts included significant manual code review and penetration testing and involved every possible technique in the process of finding vulnerabilities and verifying security. We have also been deeply involved in designing AST programs for managing AST efforts across thousands of applications.

There simply is no one right way to perform AST. A tool or technique that works well on one application could be awful on another application. An approach that works well for one project could be awful on a project with a different development process. This variability is because every application has:

- A unique set of threats, requirements, and regulations
- A unique architecture, framework, libraries, and security defenses
- A custom development process and pipeline

Many organizations have hundreds or thousands of applications that need testing. The goal of new NIST standards, procedures, or criteria for AST should be to drive the highest degree of assurance in the most cost-effective manner across a large and rapidly changing application portfolio. To achieve this, we recommend the following:

1. **Encourage focused AST**. NIST should ensure that all AST results include traceability back to a threat model identifying the most critical risks. Most AST efforts are wasteful and duplicative – often testing for vulnerabilities that are not possible for a given application – such as testing for SQL Injection in an app with no database. Or blindly running an AST tool without considering what it tests for or how well it works. NIST would do the world a great service by enabling teams to choose the best approach for each item to test, rather than simply forcing them to run multiple tools with overlapping rulesets and wildly different strengths.

2. **Encourage IAST**. NIST should encourage the use of IAST. Interactive Application Security Testing (IAST) is a widely used, fully-automated AST approach that leverages instrumentation to test applications from within. This “inside out” approach has speed, accuracy, coverage, and scalability advantages over SAST and DAST, and integrates very well with modern software pipelines. IAST is already included in NIST 800-53 rev. 5 requirement SA-11(9) and in the PCI Sofware Security Standard requirement 10.2.

3. **Disclose all vulnerabilities**. NIST should mandate that any vulnerabilities discovered that affected production applications at any time must be disclosed, even if never attacked or breached. Full visibility into development security not only informs the public and oversight officials about the security of government software, but creates a culture of "security in sunshine" that encourages teams to produce more secure software.

4. **Encourage remediation**. NIST should create incentives for remediating vulnerabilities, not simply finding them and adding them to a risk register. AST accomplishes nothing if vulnerabilities are not remediated. Therefore, AST programs should be designed to encourage remediation, not simply run a lot of tools or generate a lot of findings. We have extensive data showing that continuous testing and realtime feedback, as opposed to periodic scanning and PDF reports, lead to dramatically reduced security backlogs and MTTR under a week.

5. **Enable teams to choose a set of AST approaches**. NIST should ensure that teams are empowered to choose the best AST approach for each test, depending on the application. Each AST technique has different strengths and weaknesses both overall and for each "rule" depending on the application. Both the NIST SAMATE and OWASP Benchmark results provide ample evidence here. We recommend that NIST encourage teams to use AST tools surgically for what works well on their application. NIST should also require teams to justify their choices by explaining why each strategy is a thorough and accurate test.

6. **Encourage automation**. NIST should create incentives for organizations to automate continuous AST. Modern applications are tremendously large and complex and are being developed at amazing speed, often releasing multiple times a day. Automation is absolutely critical for maintaining assurance at scale. Activities that require significant human expertise, like code review, penetration testing, and static analysis, should be considered security research that adds to the tests performed by fully automated AST tools.