REAL-TIME HYBRID ANALYSIS: FIND MORE, FIX FASTER

Technology white paper

Brian Chess, Ph.D., Distinguished Technologist, HP
Founder and Chief Scientist, HP Fortify

Summary

Real-time hybrid analysis marks a substantial evolution in software security testing. It provides unique access to application information missing from the two most effective software security analysis technologies in use today—dynamic and static testing. This insight enables real-time hybrid analysis to overcome the shortfalls that have thus far limited the effectiveness of applying these methods in combination. Using real-time hybrid analysis, organizations can analyze software with far greater thoroughness, precision, and efficiency than previously possible to identify more vulnerabilities, improve the accuracy of diagnosis, speed remediation efforts, and simplify software security processes.

ENTERPRISE SECURITY

A vulnerability glut

The exponential growth of software applications and their ubiquitous accessibility make security a daunting endeavor for even the best funded and staffed IT organizations. As high-profile security breaches involving Sony, Citigroup, and legions of others demonstrate, exploitable vulnerabilities in software introduce substantial risk. While the sheer number of applications continues to soar, so does the prevalence of vulnerabilities and the severe repercussions caused by insecure software. Compounding the problem is the complexity of modern software, which increasingly targets versatile, "always-on" scenarios including Web 2.0, mobile, and the cloud.

Against this backdrop, software security practitioners and developers, facing business mandates for efficiency and profitability, are often compelled to secure applications more rapidly while using fewer resources. Making the task yet more difficult is the labor-intensive nature of software security assurance processes. To successfully distinguish critical vulnerabilities that must truly be addressed from those that involve little to no risk can require substantial effort, far beyond the capacity of most IT organizations. Understandably, solutions that can automate the most arduous software security tasks have generated great interest in recent years. Among available candidates, hybrid technology has been perhaps the most compelling.

The foundations of hybrid analysis: Dynamic and static testing

The most effective automated vulnerability detection techniques available today are Dynamic Application Security Testing (DAST) and Static Application Security Testing (SAST). DAST works by attacking the application under test using techniques akin to those a hacker might employ. It tries many attack scenarios and monitors the application's response in order to diagnose vulnerabilities. SAST (also known as source code or binary analysis) finds security vulnerabilities by examining software without executing it.

Pros and cons of DAST and SAST

DAST and SAST each possess unique strengths. DAST is ideal for conducting an end-to-end system test. In just minutes, it can attempt thousands of attacks against an application, whether staged or in production. It automatically discovers application entry points (also known as the attack surface) and delivers attack payloads from an extensive knowledgebase. SAST is comprehensive in nature (it simulates all possible outcomes and inspects every line of code) and can identify more types of vulnerabilities than any other analysis method. Additionally, SAST provides full root-cause analysis, which pinpoints the location of vulnerabilities with line-of-code precision.

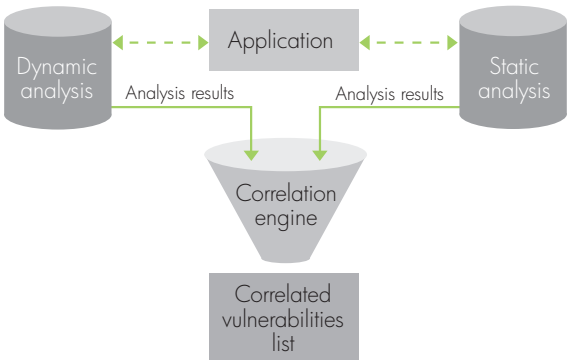
However, each method also has its weaknesses. DAST must explore the attack surface to launch a successful attack, but its knowledge of potential attack pathways is sometimes incomplete, inhibiting its ability to fully test an application. Additionally, DAST is able to detect only the symptoms of a vulnerability, not its underlying cause within the code. DAST also cannot observe an application's internal behavior. For example, if a DAST tool launched a successful SQL injection attack that destroyed a database, the only symptom DAST might detect would be the appearance in HTTP of a "404 – Page Not Found" error message, with no insight into the error's cause. In this scenario, and others like it, DAST might register the attack as meaningless or even unsuccessful, and hence the underlying vulnerability would slip through undiagnosed. And while SAST offers greater coverage and is extremely proficient at finding potential vulnerabilities in source code, it does not produce concrete test cases to demonstrate the exploitability of the vulnerabilities it finds.

First-generation hybrid analysis: A vital first step that doesn't go far enough

The allure of hybrid analysis is obvious: Combining the results from DAST and SAST holds the potential to maximize the advantages of each—the vulnerability substantiation of dynamic testing with the application coverage, root-cause analysis, and line-of-code specifics of static testing.

The first hybrid analysis tools were introduced just a few years ago. They help enterprises conduct more complete security testing, validate results through enhanced correlation, and reduce the time and expense of resolving application security issues. And yet they do not go far enough when it comes to realizing the full potential of hybrid analysis. One key reason why is because first-generation hybrid works by correlating results *only after* testing is complete (Figure 1).

Figure 1: During first-generation hybrid analysis, DAST and SAST have limited information about application behavior during a test, and correlation takes place only after testing is complete. Additionally, correlated vulnerabilities are the primary output of testing.



Room for improvement

One of the limitations of first-generation hybrid is that because vulnerability correlation happens after attacks have occurred and testing is concluded, important opportunities for more thorough analysis can be missed.

Another issue is that it can be difficult to readily align the results of DAST and SAST analysis because the two technologies process two very different types of information under very different circumstances. DAST examines web traffic while applications are under attack; its output is oriented around HTTP traffic. In contrast, static analysis scrutinizes source code and configuration files. Therefore, in order to match up results, the correlation algorithm must track down how a given vulnerability described within the relevant HTTP traffic by DAST links to a specific line of code or configuration file identified through SAST. This correlation can be difficult to perform accurately, which undercuts the ability of hybrid to make more rapid remediation possible.

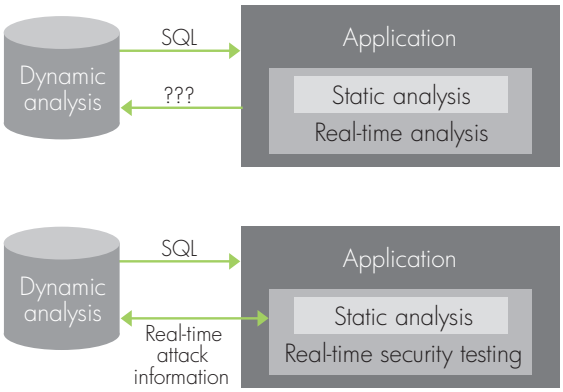
Additional concerns with first-generation hybrid involve questions of accuracy and application coverage. By focusing on vulnerabilities detected by DAST *and* SAST, and hence with a high degree of correlation between the two techniques, first-generation systems may inadvertently downplay the potential risk from vulnerabilities detectable by only one method or the other, but not by both. Moreover, as mentioned previously, because a DAST tool lacks key information about the interior landscape of an application, the attack surface it targets may be incomplete.

The missing link: Real-time application insight

A critical element missing from first-generation hybrid analysis is information about the inner workings and behavior of applications undergoing DAST and SAST analysis. This is precisely the insight provided through Real-Time Application Security Testing (RAST). Developed by HP and its subsidiary Fortify Software, RAST enables an industry first — *the ability to dramatically improve analysis results by observing applications in real time at the code level while they are being attacked and to use this information to inform and guide DAST and SAST analysis while a test is still underway.*

RAST operates in much the same way as a profiler or a debugger. Because it has the ability to see inside the process space of the running application, RAST can observe and record information about requests made to the application, the code the application executes as a result, and the values of variables inside the running program. Furthermore it can make this information available to the analyzer while an attack is taking place. RAST is similar to SAST in that it employs a collection of rules that define vulnerable behavior in terms of code-level interfaces, and yet it also has DAST's ability to observe a concrete execution of the program.

Figure 2: During real-time hybrid analysis, RAST technology feeds critical application information to analysis tools in real time while a test is taking place, dramatically improving the thoroughness and accuracy of results.



Consider the introduction of RAST to the SQL injection scenario mentioned previously (Figure 2). In this instance, RAST is able to detect that an input parameter contains SQL metacharacters. It then observes the SQL statements the application assembles, and can recognize when a malicious query is about to be delivered to the database. It communicates all of this information in real time to DAST, which is then able to capture and report the incident as a vulnerability.

Next-generation breakthrough: Real-time hybrid analysis

RAST technology provides the foundation for the next generation of hybrid analysis—real-time hybrid analysis. Real-time hybrid analysis significantly enhances code coverage and accuracy, while fully automating the process of identifying, locating, organizing, and ranking the severity of vulnerabilities in code.

Using real-time hybrid analysis, organizations can resolve their most critical software security issues faster and more cost-effectively than any other available analysis technology. Key benefits include:

- **Identification of more vulnerabilities:** RAST technology enables analysis tools to investigate more of an application's attack surface because it is capable of observing application details statically and at runtime. For example, RAST conveys critical details about file systems and the contents of configuration files to enable it to target areas of code it otherwise would not have known to attack.
- **More accurate diagnosis:** RAST also enhances vulnerability diagnosis by observing code execution in response to an attack, enabling DAST to know whether an attack has succeeded and therefore represents a vulnerability.

SPOTLIGHT: RECONSIDERING HYBRID ANALYSIS

Within the software security community, much has been made of the shortcomings of first-generation hybrid analysis. Here is a sampling of common industry criticisms of the initial technology and how they no longer apply in light of RAST and real-time hybrid analysis.

- **Claim: Hybrid doesn't work because it's hard to detect attack surface with static analysis.** RAST detects attack surface not only using static analysis techniques (by examining the file system and configuration files) but also as the program is running (e.g., to determine attack surface for late-binding frameworks).
- **Claim: Correlating static and dynamic results doesn't work.** This notion refers to the difficulty of lining up disparate results from static and dynamic testing. RAST puts this claim to rest because it has the unique ability to examine HTTP requests and code simultaneously and seamlessly link them together.
- **Claim: Hybrid doesn't help with false negatives or false positives.** Real-time hybrid greatly reduces false positives and false negatives by enhancing the diagnostic capability of DAST. It minimizes false negatives by communicating attack results to DAST in real time, enabling it to expand attack surfaces dynamically while a test is ongoing, and hence catch more legitimate vulnerabilities. Real-time hybrid reduces false positives by providing internal application data to the DAST engine it previously did not have access to.
- **Claim: Hybrid creates a false sense of security.** This belief says that the main goal of hybrid analysis is to correlate and prioritize SAST vulnerabilities, a strategy that is inherently incomplete. In contrast, automated correlation is only one of multiple benefits of real-time hybrid, and perhaps less important than expanded application coverage, greater accuracy, and root-cause clustering.
- **Claim: Nobody is willing to monitor the execution of software.** The belief here is that because RAST must be running on the server with the application under test, IT organizations are resistant to adjusting their testing patterns for the sake of security. In reality, the demand for better application security is such that security and development teams are collaborating more closely than ever, and within the IT industry, the shift is well underway toward bringing software security assurance practices, such as those embodied by real-time hybrid analysis, into the software development lifecycle.

- **Faster remediation of critical issues:** By offering an unprecedented view of application behavior made possible through RAST, real-time hybrid analysis not only provides details of an attack and their relative level of impact, it also exposes a vulnerability's root cause in code. With this explicit guidance, security and development teams can rapidly address security issues.
- **Better understanding of vulnerabilities by distilling common causes:** One root cause is often responsible for generating thousands of vulnerability symptoms. Real-time hybrid analysis is able to group all symptoms (vulnerabilities) that share a common root cause, enabling teams to quickly eliminate multiple reported vulnerabilities by resolving a single underlying problem.
- **Simplified software security management:** Leveraging RAST, real-time hybrid analysis generates a single unified report combining DAST and SAST analysis that greatly simplifies management and oversight of remediation efforts, enabling teams to quickly determine which vulnerabilities to address first for their particular circumstances. The report lists all discovered vulnerabilities, organized by such traits as:
  - Impact of an exploit
  - Degree of correlation
  - Common root causes
  - Location in code

Additionally, unified reporting provides essential details about notable vulnerabilities that do not show a high degree of correlation, but that organizations may wish to investigate further.

Key advantages by role

Real-time hybrid analysis furnishes numerous advantages that address the primary concerns of those on the front lines of application security within their organizations.

Role	Concerns	Real-time hybrid benefits
Security practitioner	• Critical vulnerabilities falling through the cracks • Using limited time most efficiently to assess and diagnose issues	• Analysis of more attack surface • More accurate diagnosis of genuine vulnerabilities
Software developer	• Translating vulnerability reports into code fixes • Fixing only the most important issues with least time and effort	• Specific location of vulnerabilities in code • Vulnerabilities organized by impact and common root cause
Management	• Avoiding the creation of multiple reports with different conclusions about the same application • Inability to see the big picture; having to connect too many dots	• Unified reporting of all DAST and SAST results • Comprehensive, accurate view of risk

Conclusion

Real-time hybrid analysis fully delivers on the promise of merging the best aspects of dynamic and static testing into a tightly interwoven approach for rapidly resolving security vulnerabilities in software. It enables greater coverage and a far higher degree of accuracy than all other analysis solutions available today, including first-generation hybrid technology or dynamic and static testing conducted in isolation. Organizations taking advantage of real-time hybrid analysis are able to detect more vulnerabilities by expanding attack surfaces during testing. Additionally, real-time hybrid analysis makes it possible to speed the remediation of critical issues, gain better insight into vulnerability root causes, and simplify software security assurance processes with unified reporting that organizes results by severity, correlation, common causes, and precise code location.

Availability and implementation

Currently, real-time hybrid analysis is available only from HP and its subsidiary Fortify Software. Their solutions are offered through industry-leading dynamic analysis products such as HP WebInspect and static analysis products such as HP Fortify Static Code Analyzer (SCA), as well as through integration capabilities that make advanced correlation possible. The components of HP Fortify Real-Time Hybrid Analysis are simple to install and interact automatically with no disruption in testing regimens.

To learn more about how real-time hybrid analysis can help you secure your applications, contact your local HP or Fortify representative.

www.fortify.com

Created July 2011