CONTRAST SECURITY

TECHNICAL BRIEF UPDATED JANUARY 2017

ACCURATELY ASSESSING APPSEC WITH THE OWASP BENCHMARK PROJECT

BENCHMARKING APPSEC ACCURACY

In 2015, the Open Web Application Security Project (OWASP) Benchmark Project was created to measure the speed, coverage, and accuracy of application security products. Funded in part by the Cyber Security Division of the United States Department of Homeland Security (DHS), the Benchmark Project lets organizations freely assess products they have or are planning to use. The results of running application security products through the Benchmark demonstrate that most organizations need to revisit their application security technology choices because they are using products that are relatively inaccurate and have high false positive rates.

TRANSPARENT, SCIENTIFIC TESTING

The OWASP Benchmark Project is a set of tools that can be used to benchmark application security testing products. The Project is open and free, so organizations can use it to measure the application security products or services that they're using today or planning on using. It consists of a large number of test cases – some of which are false positives and some of which are true positives – and a set of programs that allow organizations to measure the speed, coverage, and accuracy of different application security products.

The aim of the Benchmark Project is to scientifically evaluate the capabilities of application security testing products, where they're good, and where they aren't. To accomplish that, the OWASP project team put together 21,000 test cases that cover 11 different categories of vulnerabilities (Figure 1, next page). To check for false alarms, or false positives, about half of the test cases are real vulnerabilities, and about half are not. The large number of test cases allows the Benchmark to use variants of each vulnerability type to pinpoint exactly where products are strong, and where they are not. For example, there's not just one kind of SQL injection test, there are dozens of test cases for SQL injection. That enables the Benchmark to test many possible data flows along with different sources and different sinks.

Figure 1: OWASP Benchmark Project Test Cases

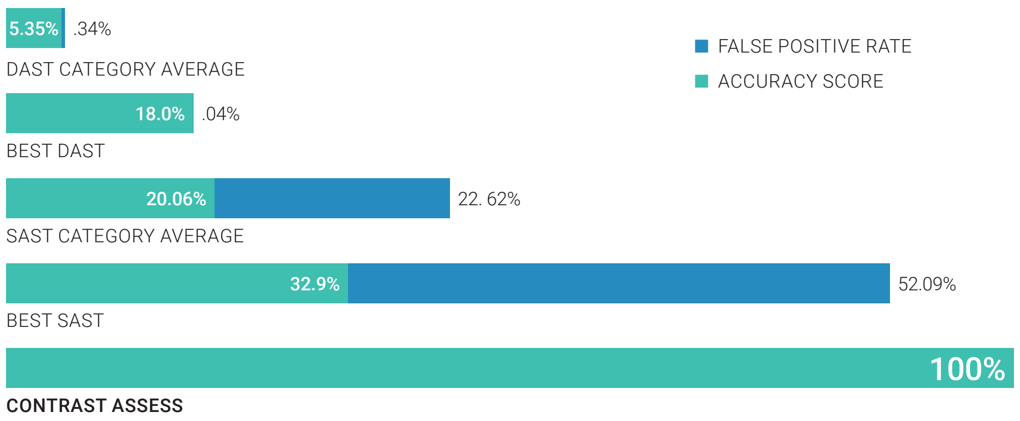
	VULNERABILITY CATEGORY	TRUE VULNERABILITIES	FALSE VULNERABILITIES
1	Command Injection	1,802	906
2	Cross Site Scripting	1,540	1,909
3	Insecure Cookie	201	215
4	LDAP Injection	521	215
5	Path Traversal	1,706	924
6	SQL Injection	2,297	1,232
7	Trust Boundary Violation	505	220
8	Weak Encryption Algorithm	720	720
9	Weak Hash Algorithm	714	707
10	Weak Random Number	1,612	2,028
11	XPath Injection	217	130
	Totals	**11,835**	**9,206**

SAST? DAST? TIME FOR IAST!

The top-level benchmark results, shown in Figure 2, are revealing and surprising. The most accurate dynamic application security testing (DAST) products had an 18% accuracy score. The most accurate static application security testing (SAST) products had a 33% score on the Benchmark. Contrast Assess, an interactive application security testing (IAST) product scored a 100%.

The Benchmark results call into question the way organizations are running their application security programs today, with such heavy reliance on SAST and DAST products. The results suggest that businesses need to look at the strengths and weaknesses of the products that they are using and seriously consider adding additional protections against the areas where their existing products are not delivering.

Figure 2: Benchmark Accuracy Results



The Benchmark Scoring Algorithm: False Positives Matter!

In Figure 2, the total column height is the percentage of all the true positives reported for that class of product or product. The red at the top of the column is the measure of false positives. The blue portion is the accuracy score, which is simply the true positive rate (total column height) minus the false positive rate (red portion). For example, Contrast scored a 100%, which comes from a 100% true positive rate minus a 0% false positive rate. The rationale behind the Benchmark accuracy score algorithm is that it takes about the same amount of time to investigate a true positive as it does a false positive. That is, every false positive that must be investigated costs the organization – in "opportunity time" – the ability to fix one real vulnerability. So, the Benchmark scoring cancels out one true positive point for every false positive point.

THE BENCHMARK EVOLVES

The OWASP Benchmark Project started out with a focus on SAST products, with over 21,000 test cases written in Java. Though relatively new, the Benchmark Project has advanced in several ways, and continues to grow as more members of the application security community participate.

One evolution was the addition of DAST coverage. To accommodate DAST products, the Benchmark team selected a subset of 2,700 tests, chosen at random, and packaged them into version 1.2beta. The smaller test suite size was necessitated because DAST products use a testing technique called "fuzzing." Fuzzing resulted in DAST products running into data storage issues as each of the 21,000+ Benchmark test cases translated into multiple DAST fuzzing tests.

Version 1.2beta also brought the addition of IAST support, which was provided with the addition of a Selenium script, included as part the Benchmark, which exercises all of the test cases in the test suite.

With SAST, DAST and IAST product coverage now part of the Benchmark, the Project team plans to extend coverage to Runtime Application Self-Protection (RASP), and Web Application Firewall (WAF) products as well.

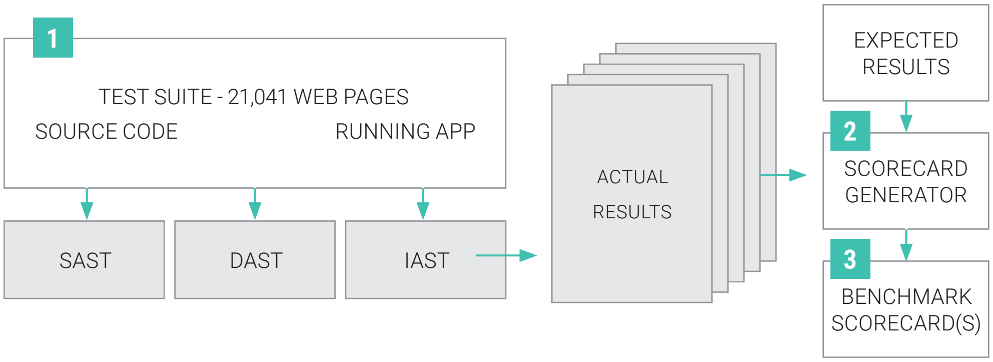
And, while the Benchmark test cases are written in Java, the Benchmark team is looking for community members to help expand the Benchmark concept to .NET, PHP, Ruby, etc.

A Project roadmap can be found on the OWASP Benchmark website at:
https://www.owasp.org/index.php/Benchmark#tab=RoadMap

GET IT ON GITHUB

The Benchmark exists as a GitHub repository (*https://github.com/OWASP/benchmark*) that can be cloned for local use. Once cloned, organizations may run their application security testing products against the Benchmark's test suite. The test suite consists of test cases in the form of Java source code that can be analyzed as-is using SAST products, and can be executed for use in testing DAST and IAST products. Figure 3 provides an overview of the Benchmark structure and process.

Figure 3: Benchmark Structure and Process



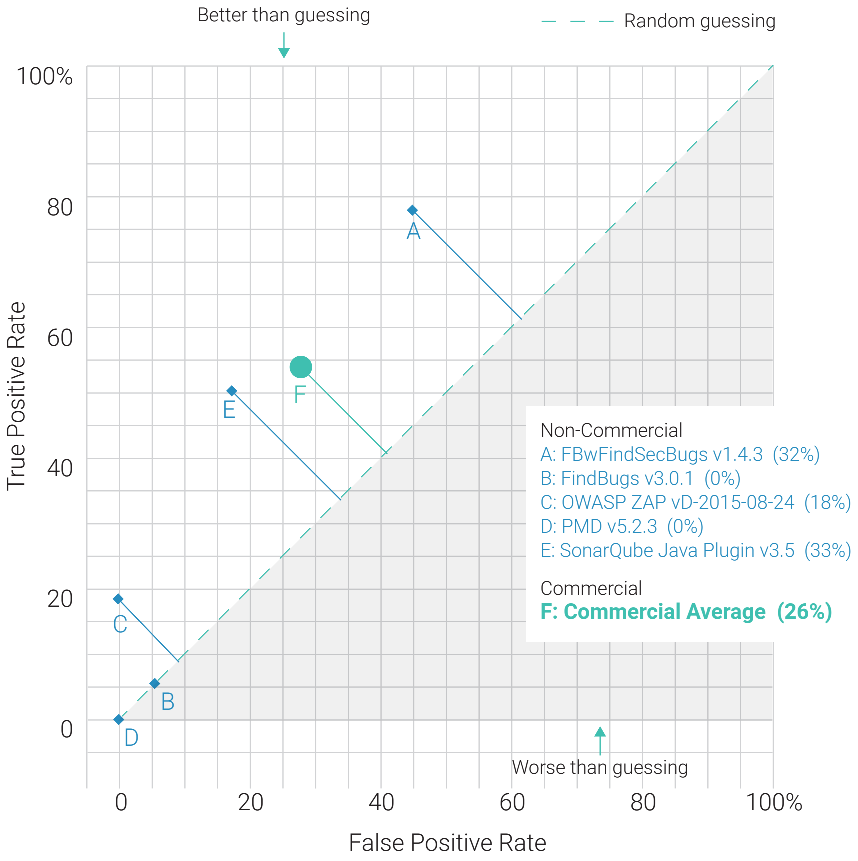
The output of the product tested is fed into the Benchmark's Scorecard Generator tool, also part of the GitHub repository. The Scorecard Generator takes the vulnerability analysis results of the product under test and compares it to the expected results for the test cases. Based on the actual versus expected results, the Generator then creates a detailed HTML scorecard that has some visualizations of the data and also a detailed chart with all the data in it. That scorecard allows businesses to see where products are good and bad, and it also helps product vendors and authors by telling them where their products are weak, so they may focus on improvements.

The Benchmark makes it very easy to test products and includes scripts to generate some basic results for open source products directly from within the Benchmark environment (Figure 4 below provides example output from that). The GitHub repository also includes directions for running commercial products and feeding their results to the Benchmark Scorecard Generator.

TRACKING FALSE POSITIVES, AND MORE

While Figure 2 above provides the overall results in a traditional column chart, the OWASP Benchmark Project Scorecard uses a chart format called a "Receiver Operating Characteristic" chart, shown in Figure 4. The results shown in Figure 4 are from testing a number of open-source SAST tools with the Benchmark test suite, and comparing them to the average Benchmark score for commercial SAST products.

Figure 4: OWASP Benchmark Scorecard



Receiver Operating Characteristic charts are a standard way of reporting results for tests in environments where true positive and false positive rates are important, such as medical diagnostics and application security.

In a Receiver Operating Characteristic chart, an ideal accuracy score would be in the upper left-hand corner. That's a score with a very high true positive rate and a very low false positive rate.

A score in the lower left-hand corner means a low true positive rate, but also a low false positive rate. In terms of application security vulnerability testing, that would be a product that reports almost no vulnerabilities. An extreme version of this would be a relatively trivial product to create: it would just report "no vulnerabilities found" for any application tested!

Products scoring in the upper right corner are more interesting because they have a high true positive rate. Unfortunately, they also have high false positive rates as well. Again, the extreme case of this would be a relatively easy product to create: it would just need to report "vulnerability found" for everything it tested. Yes, it would flag every vulnerability, but it would have a 100% false positive rating because it claims that every totally bug-free part of the application has a security vulnerability, when in fact there is none.

The diagonal line that goes across the chart from the lower left to the upper right is what OWASP calls "the random guess line." Products with accuracy scores on that line, or close to that line, aren't providing a whole lot of value. And, if a product's score is in the middle of that line, it's just like flipping a coin on whether there's a vulnerability or not.

No product should score below the random guess line, because then it performs worse than random guessing, which doesn't make a lot of sense. One could just do the opposite of what that product reported and be better off!

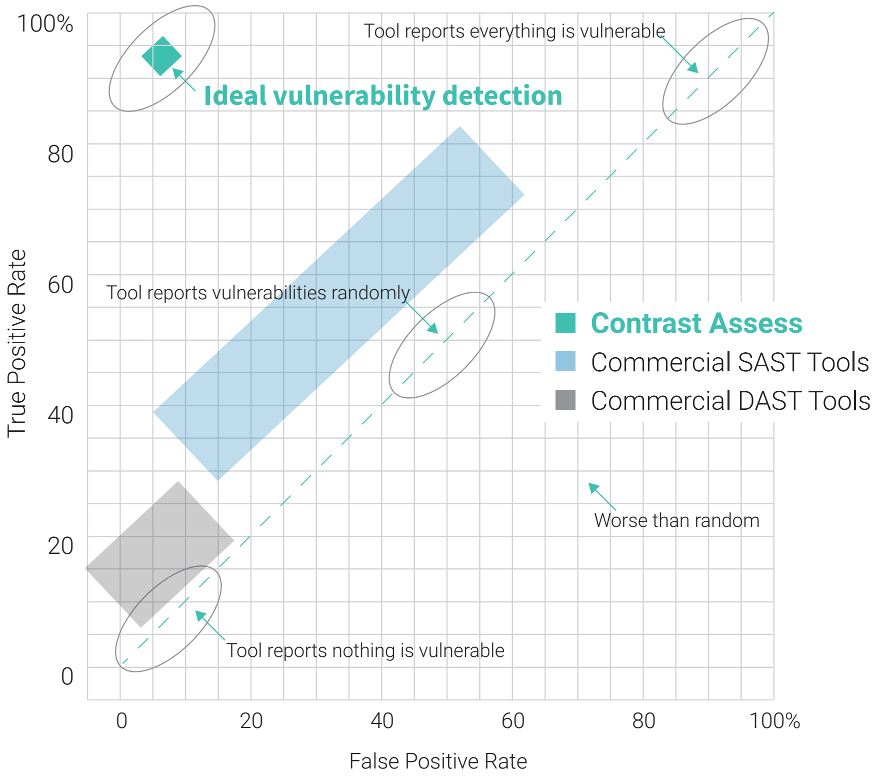
RESULTS CLUSTER

Figure 5 (next page) shows the results of the DAST and SAST products tested, as well as Contrast Assess. The scores for DAST products generally clustered in the lower left of the OWASP Benchmark Scorecard. They ran with a relatively low false positive rate, but they didn't find a whole lot of true positives either. In other words, what they found was relatively good, but they didn't find a lot of vulnerabilities.

SAST product results fell into a broader range. At the low end, SAST products got relatively low false positives, around 15-20%, which is fairly accurate; but their true positive rates, about 30-40%, were also relatively low. At the other end of the SAST cluster were products that identify more true positives but also lots of false positives – nearly 60% in some cases.

This wide variation in product results demonstrates why it's critical to understand how categories of products and individual products perform.

Figure 5: Benchmark Result Clusters



BENCHMARK SUPPORTS BROAD PRODUCT RANGE

The OWASP Benchmark Project can generate results for the products – open source and commercial – listed in Figure 6. The Benchmark Project has reported detailed results for the open-source products, but commercial vendor results are only available for vendors who agree to share that data via the Benchmark Project. Until commercial vendors opt-in, the Benchmark Project team is reporting aggregate results (e.g., those in Figure 2 and Figure 5).

Figure 6: Products supported by the OWASP Benchmark Project (as of October 2015)

ACUNETIX WEB VULNERABILITY SCANNER	ARACHNI	BURP PRO
CHECKMARX CXSAST	CONTRAST ASSESS	COVERITY CODE ADVISOR
FINDBUGS	FINDBUGS WITH THE FINDSECURITYBUGS PLUGIN	HP FORTIFY
HP WEBINSPECT	IBM APPSCAN	IBM APPSCAN SOURCE
OWASP ZAQP	PARASOFT JTEST	PMD
RAPID7 APPSPIDER	SONARQUBE	VERACODE SAST

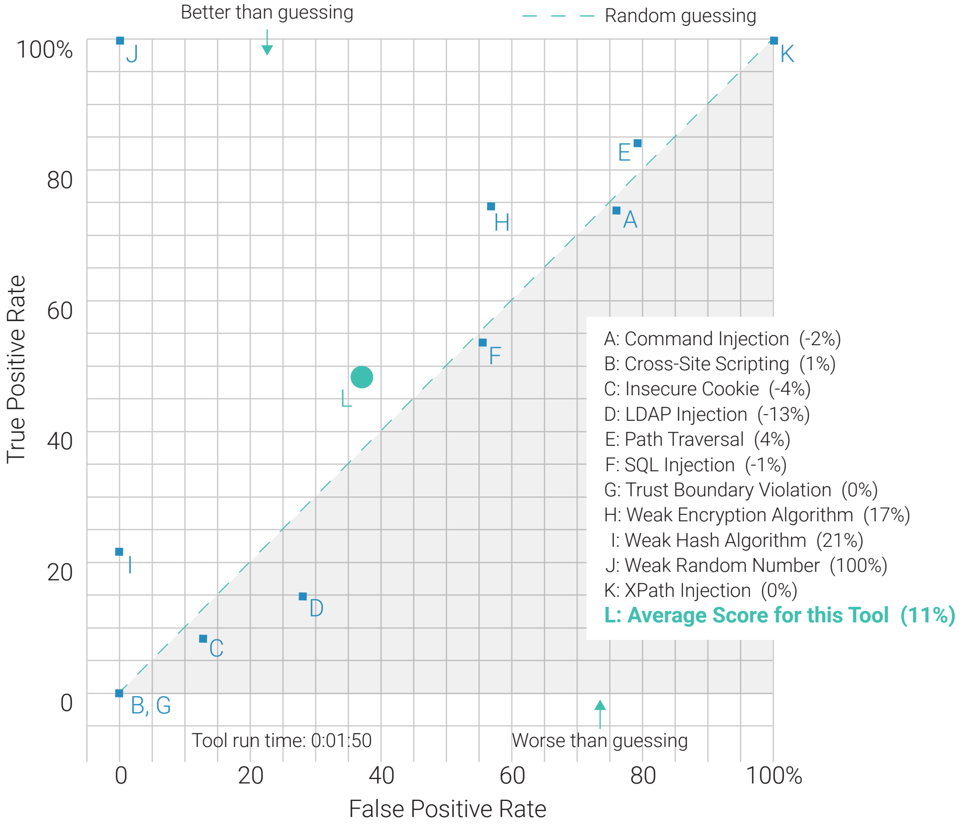
The OWASP Benchmark Project team has been running most products “out of the box” and then tuning them as necessary to make them work with the Benchmark. They are partnering with vendors to create procedures that make it easy for anybody to run products with the Benchmark.

To-date, Contrast Security has agreed for its Benchmark results to be shared, and HP has published some results in a recent Dark Reading editorial post. Veracode has written a blog post endorsing the concept of a benchmark, but has not published their results. Businesses can use the Benchmark Project to test commercial products themselves and can ask application security vendors about their OWASP Benchmark accuracy score (note: while vendors may self-report results, results obtained from the Benchmark Project are independently validated).

DRILL DOWN FOR MORE INSIGHT

Benchmark Scorecards for individual products reveal more detail than accuracy alone, as Figure 7 shows. The Benchmark demonstrates that different products perform differently against different kinds of vulnerabilities. For example, some products may be good at identifying LDAP injections, while other products of that same type (i.e., SAST, DAST, IAST) might not find those kinds of vulnerabilities at all. Figure 7 highlights this for the open source SAST product “Find Bugs,” with the “Find Security Bugs” plug-in. Yes, it is perfect at finding Weak Random Numbers (point “J”), but it falls terribly short in most of the other categories.

Figure 7: Benchmark v1.2beta Scorecard for FBwFindSecBugs



So, it's important to understand which products produce great results, in which areas, so that you may confidently pass those along directly to developers to fix. Similarly, it's critical to know which products generate lots of "noise" around certain vulnerabilities, which may then require expert assistance. And it's important to know which application security products run in real time and provide great results to processes like Agile and DevOps. All of these factors should be factored in to the product selection process to ensure that the products are really supporting business and security goals, and not just taking on a life of their own.

TIME TO REEVALUATE APPLICATION SECURITY PRODUCTS AND PROGRAMS

With the OWASP Benchmark, organizations now have a way to systematically evaluate the strengths and weaknesses of their current solutions and alternatives. Contrast Enterprise, which the OWASP Benchmark demonstrated is exceptionally accurate, is a natural choice to augment or replace existing SAST and DAST solutions. Ask your application security vendor for their Benchmark results, and contact Contrast Security (*benchmark@contrastsecurity.com*) to learn more about Contrast Assess.

APPENDIX: CONTRAST SECURITY, JEFF WILLIAMS, AND THE BENCHMARK

Two facts related to the OWASP Benchmark Project makes one curious about how Contrast Security is involved in the Project. The first is that Contrast Assess, the Contrast Security flagship product, scored the best of any product tested to-date with the OWASP Benchmark Project test suite. The second is that Jeff Williams, Co-Founder and Chief Technology Officer of Contrast Security helped run the OWASP organization for a number of years.

So, as part of an October, 2015 Contrast Security webinar, Jeff Williams was asked about those things. The following is excerpted from that webinar, and the entire webinar is available for viewing at: *http://www.contrastsecurity.com/ondemandowaspbenchmark1015*

Moderator: "…[A] lot of people associate you with OWASP. So what's been your involvement with this benchmark?"

Jeff Williams: "I really haven't been part of the OWASP organization since I stepped down as Global Chair in 2012, after running OWASP for eight years. OWASP has several hundred open-source projects, and I've been involved with a few of the most successful ones over the years, like the OWASP Top 10 and WebGoat and ESAPI. But this Benchmark Project is run by a guy named Dave Wichers, who's been working hands-on with App Security and AppSec products for over 15 years, and he's always been seeking better ways to help his clients select products and to do his work. I know Dave really well. We started Aspect [Security] together. We ran OWASP together for many years. Now that I'm at Contrast, I rarely see him. I love talking about the Benchmark with him because for AppSec geeks, it's really exciting."

Moderator: "So the Benchmark then is just like any other OWASP project…It's open source. Anyone can use the Benchmark, see how it's constructed, comment on it, and contribute to it?"

Jeff Williams: "Exactly, and there have been numerous people that have contributed, including many open-source tool authors. Some commercial vendors have been participating in the project and contributing both effort and code into the project. The key here is that the project is completely transparent, so anybody can look at the test cases, anybody can reproduce the results. And the hallmark of good science is control. You have to be able to reproduce those results, and that's what this benchmark enables."

Moderator: "Obviously we're doing this webinar because we did really well. We scored in that sweet spot. How would you explain how Contrast [Assess] did so well on this? It's pretty astounding. What are the one or two things that really make us so much more accurate, with so far fewer false positives?

Jeff Williams: "Sure. Well, fundamentally finding vulnerabilities is all about having all the right information that will allow you to determine whether there's a vulnerability, accurately, or not. So, static analysis tools attempt to do that by only looking at the source code, which doesn't have information about the HTTP requests and the back-end connections and things like that. Dynamic tools try to do it only by looking at HTTP requests and responses. And, that's really not enough information to see a lot of security vulnerabilities. But Contrast [Enterprise] works from inside your running application.

Again, with Contrast, you take our agent, you drop it on your application servers, whether that's dev or a test or staging. And Contrast [Assess] instruments the application with sensors that allow it to see what's going on from inside the running application. So it can see not only the code and the HTTP requests, but it can also see the full run time data flow. It can see the libraries and frameworks that are in use. It can see configuration files. It can see the actual backend connections and it just has a lot more information that allows it to be extremely accurate when it's identifying vulnerabilities. Contrast [Assess] actually watches the vulnerable behavior occur in the application and only reports things that actually happen. That's why it's got such a low false positive rate. Because it's not guessing, it's only reporting things that actually happen and are provably vulnerable."

CONTRAST SECURITY

240 3rd Street
Los Altos, CA 94022
888.371.1333

120616