



# Good, Better, Best Software Testing

If you're testing your software, you're already one step ahead of the bad guys.

Most of the time software testing means static analysis security testing (SAST). This is a form of white box testing where the tool has access to the source code. It represents the developers point of view. It identifies known weaknesses that could become exploitable vulnerabilities later and suggests ways to remediate those weaknesses.

More mature software testing involves the use of software composition analysis (SCA). This is a form of black box testing where the tool has access to both the source code and the binary, which may be from a third party. It identifies known vulnerabilities in third-party components and suggests upgrades, patches, or workarounds.

Advanced Fuzz Testing (AFT) addresses the larger question of unknown unknowns, the space where zero day vulnerabilities -- vulnerabilities that lie dormant waiting to be found -- live. It does not rely solely on lists of known software weakness or vulnerabilities alone. It executes across a broad range of code to find new vulnerabilities, then tests against those to make certain they pose a risk. AFT is comparable to a popular form of manual testing known as penetration testing.

If you're not using AFT alongside your traditional software security testing, then you're missing a big part of the picture.

**SCA / Common Vulnerabilities Enumeration (CVE)**

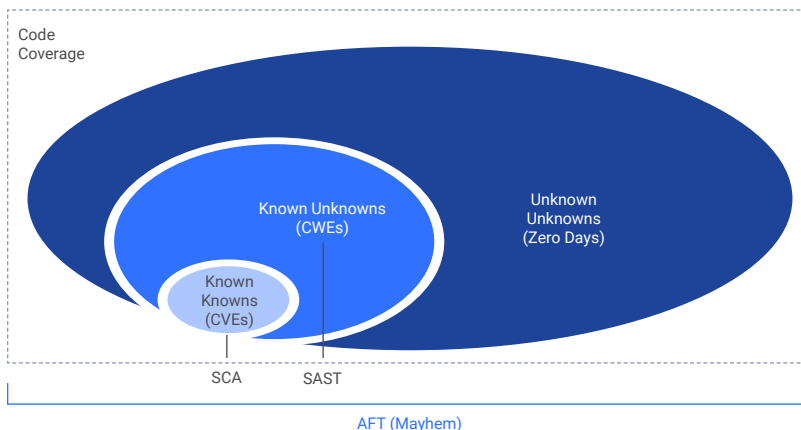
Known software vulnerabilities that are categorized and rated by criticality.

**SAST / Common Weakness Enumeration (CWE)**

Known software weaknesses that can lead to unknown software vulnerabilities.

**AFT / Zero Days**

Unknown software vulnerabilities not yet reported to the software vendor.



SAST	SCA	AFT
<p>Known Unknowns</p> <ul style="list-style-type: none"> <li>Identifies unknown SW <b>defects that could lead to a compromise (CWE)</b>.</li> <li>There are anywhere between 10,000s to 1Ms of defects in a given software</li> </ul>	<p>Known Knowns</p> <ul style="list-style-type: none"> <li>Identifiable known <b>SW defects that could lead to compromise (CVE)</b>.</li> <li>There are anywhere between 10s to 100s of CVEs that may exist in the software.</li> </ul>	<p>Unknown Unknowns</p> <ul style="list-style-type: none"> <li>Identifies unknown SW risks.</li> <li><b>There are no identifiers (i.e. CVE or CWE)</b>.</li> <li>The quantity of unknown unknown risks is unknown</li> </ul>
<p>White box security testing</p> <ul style="list-style-type: none"> <li>Tester has access to the framework or application code</li> <li>Application tested from inside out</li> <li>Represents developer's view</li> </ul>	<p>Black box security testing</p> <ul style="list-style-type: none"> <li>Tester has no knowledge how the application was created</li> <li>Application tested from the outside in</li> <li>Represents the hacker's perspective</li> </ul>	<p>Gray box security testing</p> <ul style="list-style-type: none"> <li>Tester has both access to the application and the binary</li> <li>Application tests both inside &amp; out</li> <li>Represents the developer's &amp; hacker's perspective</li> </ul>
Does check every line of code	Doesn't check every line of code	Doesn't check every line of code
High number of false positives	Low number of false positives	Low number of false positives
Manual test case creation	N/A	Autonomous test case creation
Slow interaction with code	N/A	Fast interaction with code
Requires access to source code	Requires binary code	Requires either source code or binary code
Finds vulnerabilities earlier in SDLC but not later	Finds vulnerabilities before and after software release	Finds vulnerabilities before and after software release
Less expensive to fix vulnerabilities	More expensive to fix vulnerabilities	Less expensive to fix vulnerabilities
Doesn't discover defects at run-time	N/A	Can discover defects at run-time

Want to learn more?  
 Download the [“Buyer’s Guide on Application Security Testing”](#)  
 for more details on SAST, SCA, and AFT.



ForAllSecure