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

Most of the time, software security testing means static analysis security testing (SAST). This is a form of white box testing where the tool has access to the source code similarly to 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 or patches.

Mayhem 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. Mayhem is comparable to a popular form of manual testing known as penetration testing.

If you’re not using Mayhem alongside your traditional software security testing, then you’re missing a big part of the picture. For instance, Mayhem for Code autonomously searches for zero day vulnerabilities so that even your unknowns unknowns become known. At ForAllSecure, our focus on application security allows developers to focus on building new features, new software, and new tomorrows.
SAST: Known Unknowns
- Tester has access to the framework or application code
- Application tested against known CWEs
- Represents developer’s view

SCA: Known Knowns
- Tester has no knowledge how the application was created
- Application tested against known CVEs
- Represents the hacker’s perspective

Mayhem: Unknown Unknows
- Tester has access to the application
- Application tests for both known and unknown vulnerabilities
- Represents both the developer’s & hacker’s perspective

Good
- High number of false positives
- Uses predefined checkers
- Requires access to source code

Better
- Lower number of false positives
- Looks for known vulnerabilities
- Requires binary code
- Can evaluate third-party, supply chain code

Best
- Lowest number of false positives
- Autonomously creates new test cases for more code coverage
- Works with any web app or application
- Can evaluate third party, supply chain code
- Validates each vulnerability found
- Performs regression testing to make sure fixes stay fixed

Want to learn more?
Download the “Buyer’s Guide on Application Security Testing”
for more details on SAST, SCA, and Mayhem