ForAllSecure

Why Fuzz Test?

The Attack Surface is Expanding

According to Cybersecurity Ventures, the application attack surface is growing by 111 billion lines of software code every year, with newly reported zero-day exploits rising from one-per-week in 2015 to one-per-day by 2021.

Fuzzing is Proven

Teams at Google report that fuzzing finds 80% of their bugs, while the other 20% is uncovered by other forms of testing, or in production. Organizations such as Microsoft, Carnegie Mellon University, and Google have found success with their in-house fuzzing programs.

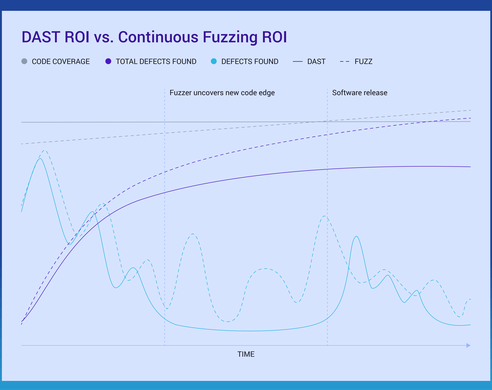
1,800
Bugs and vulnerabilities in Office
Microsoft

11,687
Bugs and vulnerabilities in Linux
Carnegie Mellon University

27,000
Bugs and vulnerabilities in Chrome and OSS
Google

Continuous Security with Continuous Returns

Continuous fuzzers are highly effective because they perpetually conduct negative testing. The Pesticide Paradox claims that if the same tests are repeated over and over again, new defects are no longer found, ROI diminishes, and defects are clustered in limited sections of the software, creating hotspots. Continous fuzzers autonomously generate new test cases for continuous ROI.



Fuzzing is Accepted

Fuzzing is a recommended practice in the Microsoft Secure Development Lifecycle (SDLC). Although fuzzing is listed under the Implementation category, "shift-left" testing philosophies states the earlier in the SDLC you can introduce it, the better.

TRAINING	REQUIREMENTS	DESIGN	IMPLEMENTATION	RELEASE	RESPONSE
Core security training	Establish security requirements	Establish design requirements	Use approved tools	Create an incident response plan	Execute incident response plan
	Create quality gate / bug bars	Perform attack surface analysis / reduction	Deprecate unsafe functions	Conduct final security review	
	Perform security and privacy risk assessment	Use threat modeling	Perform static analysis	Certify release and archive	
			ForAllSecure		

Why Doesn't Everyone Fuzz?

Until recently, fuzzing has been a software security practice exclusive to tech behemoth, such as Microsoft, Google, Amazon, Apple, and more. While the benefits of fuzzing are undeniable, it's not easy to harness its power without a commercial offering that helps organizations get started.

"Security engineers of the ClusterFuzz and OSS-Fuzz teams have disclosed that while it is possible to bootstrap and operate high-performance fuzzers, people often underestimate the complexity of upstanding such solutions. They have disclosed that even with their padded budgets and world-class experts, it took Google years to achieve full automation."

Not All Fuzzers Are Equal

Recent advancements in fuzzing have made this advanced technique available to the general public. So, what makes a great fuzzer? Listed below is a suggested buyer's criteria framework.

Efficient
Automatically and accurately uncovers defects with little time and resources

Experience
Higher CPU years indicate more experience and knowledge on a test target

Smart
Analyzes targets to generate inputs that are most likely to find defects

Reproducibility
Enable reproduction of vulnerabilities for remediation

Continuous
Perpetually tests for defects

Want to learn how fuzzing fits into your application security program?

Download the Buyer's Guide