

Don't crash
going too fast

Hacking with speed and precision

Table of contents

1.	Security testing at the speed of business	<u>3</u>
2.	Shift to the left!	<u>7</u>
3.	Give feedback quickly and often!	<u>12</u>
4.	Perform comprehensive tests	<u>16</u>
5.	Use human intelligence	<u>20</u>
6.	Break the build!	<u>24</u>



SECURITY TESTING AT THE SPEED OF BUSINESS

Technology and security should be change enablers

Security testing at the speed of business

Technology and security should be change enablers.

Businesses change so quickly and their changes shape the world. These transformations have brought better products and services to consumers while removing transactional barriers. Those who provide a differentiated convenience have an advantage and those who manage to reduce their time-to-market can enjoy an even higher competitive position. The key player in this evolution is *information security*. Technology is and will keep being a change enabler.

Security testing at the speed of business

Inevitably, technology can pose risks while making companies more competitive. In this ebook, **we show how an organization can perform its system testing without compromising quality and while doing so at high speed.**

A successful strategy must include carrying out comprehensive tests across their systems without relying only on automated technology. As technology cannot beat human judgment, nor can it project the impact of vulnerabilities on the environment in which it interacts, skilled professionals in security testing are and will remain key to guaranteeing completeness and quality. Testing processes must coordinate all stakeholders successfully to keep business thriving. Moreover, stakeholders must be empowered to contribute to these processes whenever needed.

Security testing at the speed of business

Another key methodological consideration, in line with frequent and quick feedback to stakeholders, is to perform tests from the early stages of development (known as shift-left). The sooner bugs and vulnerabilities are found, the better, as they are easier to fix and at a lower cost.

Finally, organizations can leverage the power of new technologies to a reasonable level, for instance, by using machine learning to make cybersecurity people and operations more efficient. For example, at Fluid Attacks, we have discovered a novel application of machine learning to speed up the comprehensive security testing performed by our hackers. A triage in the code audit process we perform allows us to report weaknesses more quickly. Furthermore, as in many aspects of human life, some devices or tactics are successful in preventing us from making bad choices; we suggest corporations break the build when vulnerabilities are still present in their systems.

“We show how companies can perform their systems testing without compromising quality and doing so at high speed.”

we hack your software

SHIFT TO THE LEFT!

Address the security process from the early stages

Shift to the left!

Address the security process from the early stages.

In the 70s, Barry Boehm was one of the first authors to estimate the costs of repairing software defects.¹ **In short, the costs of fixing are higher when defects are addressed in later stages.** In the 2000s, the assertions over the costs of fixing errors were similar, although other authors also contributed to new estimations.

¹ Boehm, B. (1976) Software Engineering. *IEEE Transactions on Computers*, C-25(12), 1226–1241.
Boehm, B. (1981) Software Engineering Economics. Prentice-Hall. Englewood Cliffs, NJ.

Shift to the left!

The lesson from the above is clear: **detecting and fixing defects at the early stages avoids losses.** That's what we mean by 'shift to the left.' In a sequential SDLC (Software Development Lifecycle), the testing stage is usually located to the right side. So, testing should be shifted to the left to reduce the costs of the software development process.



*Detecting and fixing
defects at the early
stages avoids
losses.*

Shift to the left!

The consequences of software defects and security breaches make brands and corporations appear on TV news and newspaper headlines. It is less known what happens to small to medium-sized businesses (SMBs). These companies are probably in a significantly weaker position to face the consequences of a security disruption.³ A shift-left approach is a must for them. The future value of shifting to the left outweighs its costs for all organizations.

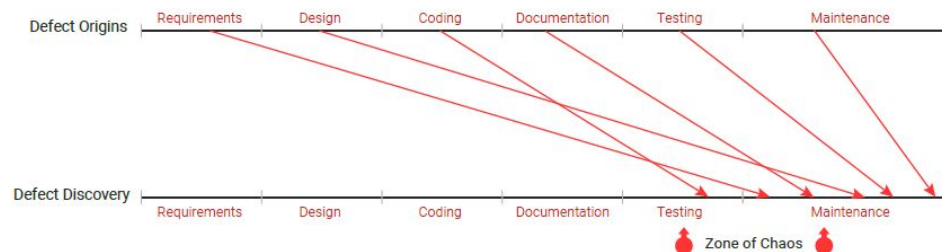


Figure 1 - Delays between defect creation and discovery when testing is the primary removal method²

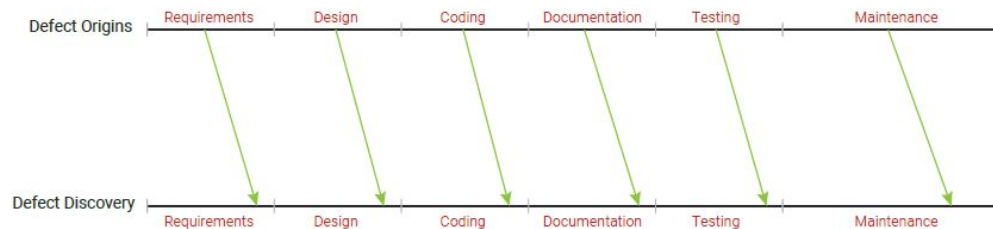
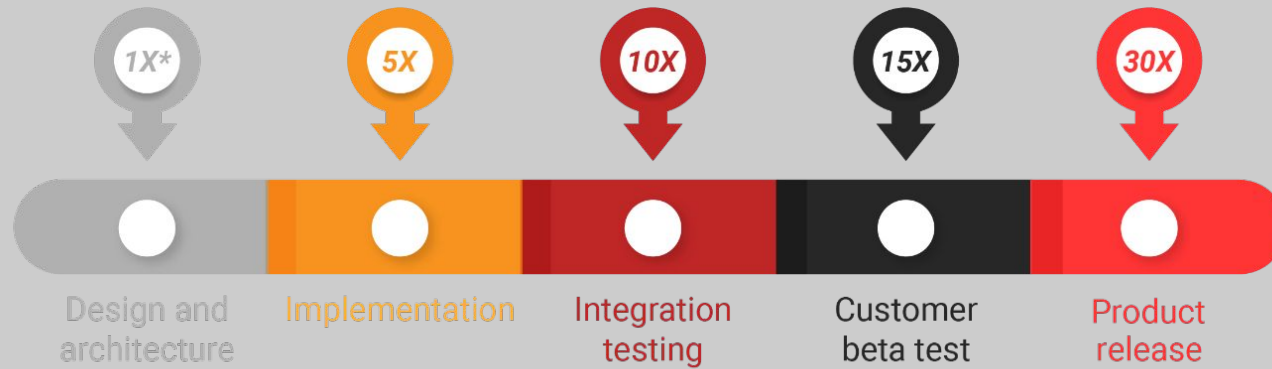


Figure 2 - Reduced delays between defect creation and discovery associated with formal inspections²

² Jones, C. (2008) Applied software measurements. 3rd ed. McGraw-Hill.

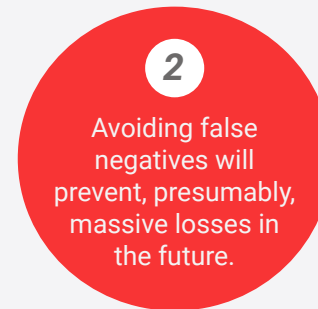
³ Bankrupt by Wrong Cybersecurity! - <https://fluidattacks.com/web/blog/smbs-bankruptcy/>



* "X is a normalized unit of cost and can be expressed in terms of person-hours, dollars, etc."

Source: National Institute of Standards and Technology (NIST)⁴

The broad benefits of investing in early and frequent technology testing are two-fold:



⁴ IBM (2008) Minimizing code defects to improve software quality and lower development costs. Development solutions White paper. Page 2.
<ftp://ftp.software.ibm.com/software/rational/info/do-more/RAW14109USEN.pdf>



GIVE FEEDBACK QUICKLY AND OFTEN!

Use an attack surface management system to
coordinate all stakeholders and give feedback
quickly

Give feedback quickly and often!

Use an attack surface management system to coordinate all stakeholders and give feedback quickly.

The world has adopted software for nearly every task imaginable, and IT in corporations has become more complex. As a consequence, vulnerabilities have also become widespread. Material and reputational losses caused by unfixed and undetected vulnerabilities led companies to reject the traditional approach to vulnerability or attack surface management: a list of vulnerabilities either in technical reports or in vulnerability scanning solutions. The speed to delivery of insights (real value) and scalability were not meeting the evolving demands of the world.

Give feedback quickly and often!

Attack surface management needed to evolve to ease all these pains and to provide more precise information in response to the complexity posed by software and systems. The market moved from isolated and often extremely slow attack surface management processes to Attack Surface Management (ASM) systems. These ASM systems consolidate bugs and vulnerabilities from different sources and provide insights from analytics tools to facilitate decision-making.



Innovative ASM programs allow companies to achieve better results by leveraging on the consolidation of defects and vulnerabilities across the software development lifecycle.

Give feedback quickly and often!

Innovative ASM programs **allow companies to achieve better results by leveraging on the consolidation of defects and vulnerabilities across the software development lifecycle**, coordinating all stakeholders, communicating timely about the progress made, and simplifying the workflow needed to maintain ongoing high-quality delivery.

Moreover, ASM programs allow stakeholders to consult the information they need whenever they need it. Project status indicators and dashboards can be created and customized to summarize the information. Details of projects, bugs, vulnerabilities and more can be accessed whenever is needed. ASM programs can also help quickly prioritize what needs to be addressed given certain conditions like severity, components and deadlines, among others. Finally, these solutions nudge stakeholders towards the most relevant action: **fixing vulnerabilities**.



PERFORM COMPREHENSIVE TESTS

Assess each technology component continuously

Assess each technology component continuously.

Information technologies can have vulnerabilities at different levels, and no single testing technique can discover all of them. What is more, finding vulnerabilities once a system is wholly coupled is often more costly and time-consuming.⁵

For these reasons, one of the suggested practices in security testing is to perform diverse methods in different moments. It is sensible to focus on different technology components, as well as in each of the stages of the development lifecycle, as the interaction of two or more components can also create vulnerabilities. Moreover, technologies also interact with one or several world environments outside of their realms (people, markets, regulations, etc.) that, in turn, can develop weaknesses that must be addressed in order to safeguard operations and information.

In the following lines, we briefly describe five methodologies we perform in our approach to security testing.

⁵ See Section 2 (Shift to the left!) for more details.

Static Application Security Testing –SAST–

Focuses on analyzing application source code, byte code, and binaries for coding guidelines, standards, and design conditions that might point to security vulnerabilities. SAST is a white-box approach.

Dynamic Application Security Testing –DAST–

Is different and complementary to SAST, as it is a black-box and gray-box testing approach (portions of the target are hidden, like the source code) and requires the application to be running. Standard DAST solutions cover only web applications.

Software Composition Analysis –SCA–

Is an automated way of identifying potential vulnerabilities given the use of third-party and open-source software and hardware components in a system.

Penetration Testing —PT—

Goes beyond previous techniques to use diverse attack scenarios that first find vulnerabilities and then attempt to exploit them. This way, a more realistic picture of the security state of a system can be achieved.

Automated Vulnerability Discovery —AVD—

Uses tools that automatically detect vulnerabilities. We recommend this for finding deterministic vulnerabilities and complement it with other approaches to detect more complicated vulnerabilities. Studies have shown consistently that the capabilities of AVD solutions are limited due to the high rates of false positives and negatives. When companies only implement AVD, they will face the issue of wasted time. Developers must confirm which are the real vulnerabilities. However, the highest risk companies face is that of false negatives: what vulnerabilities exist that AVD solutions are not capable of reporting?



USE HUMAN INTELLIGENCE

Expand the depth of testing and shrink the noise: let human judgment and intelligence bring precision and accuracy to testing by identifying undetectable vulnerabilities and getting rid of false positives

Expand the depth of testing and shrink the noise: let human judgment and intelligence bring precision and accuracy to testing by identifying undetectable⁶ vulnerabilities and getting rid of false positives.

Automation keeps growing everywhere. Nevertheless, it is far from replacing human judgment and decision-making. If we want to use these technologies to improve an organization's bottom line, we need to understand their limits.

⁶ "Undetectable" by AVDs.

In cybersecurity testing, a portion of what is relevant can be handled by automation —and we should use it. However, when left to their own devices, AVDs yield a lot of noise: plenty of false positives that can reduce operational efficiency. To overcome this, security testers must kick in. Human judgment is essential in reducing the burden of ill-identified weaknesses, thus significantly improving the accuracy and precision of testing results. Furthermore, it has been shown that AVDs alone fail to detect all vulnerabilities present in a system.⁷ That's the other side of the coin. Again, human intelligence is critical in avoiding to move a system into production that has undetected flaws. False negatives are even more problematic. These are like blind spots, in the sense that although you cannot see them, they can potentially cause massive outages or losses.⁸

⁷ An internal 3-year long experiment showed that six popular open-source AVDs failed to detect between 82% and 99% present vulnerabilities in a system. A subset of commercial AVDs did worse on average.

⁸ Two prominent examples are the CapitalOne <https://edition.cnn.com/2019/07/29/business/capital-one-data-breach/index.html> and the Marriot hacks <https://www.wired.com/story/marriott-hacked-yes-again-2020/>

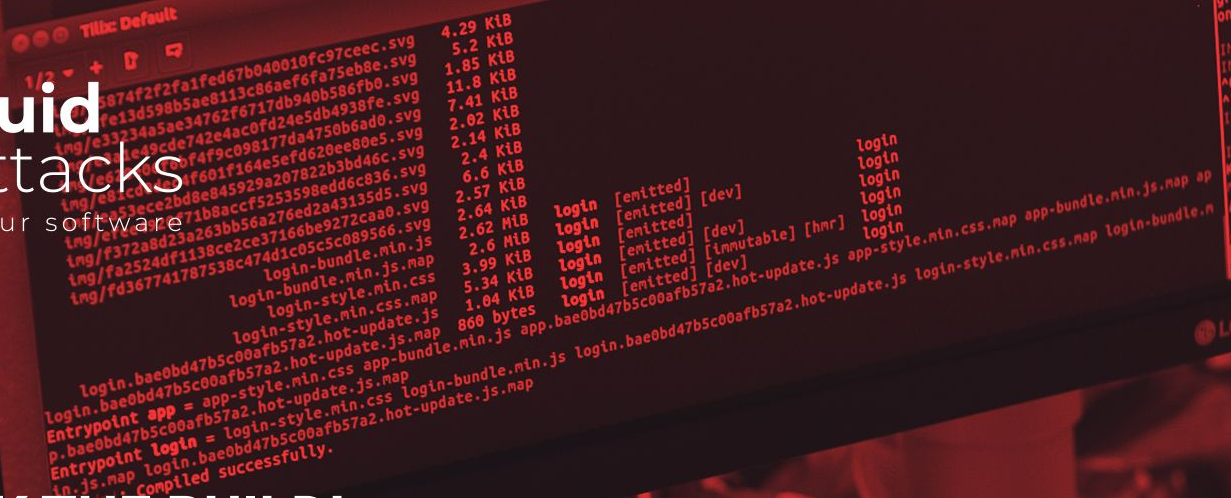
Some scattered voices in cybersecurity argue that traditional penetration testing is dead, believing that the role people play has become obsolete. We disagree. Human skills and intelligence make the most sense in any testing approach. A security testing plan that downplays the role of people can cause a lot of harm, given the proven rates of false negatives and positives. We acknowledge that some approaches relying on more automation for red teaming, like Artificial Intelligence, could be a thing, but not in the short-term. In contrast, we believe penetration testing has evolved by integrating available technology and developing better tools to enhance testing capabilities, thereby increasing the chances of spotting otherwise undetectable vulnerabilities.

“After gathering data and outputs from automated technology, hackers know better what to do and what not to do.”



BREAK THE BUILD!

Use technology to deploy an automated control in the continuous integrator that forces you to fix vulnerabilities



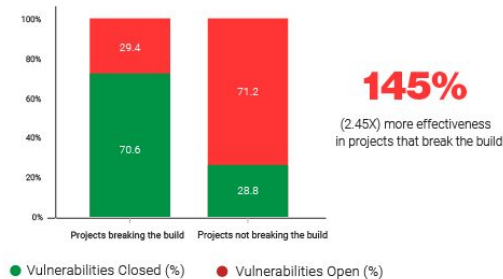
Break the build!

Use technology to deploy an automated control in the continuous integrator that forces you to fix vulnerabilities.

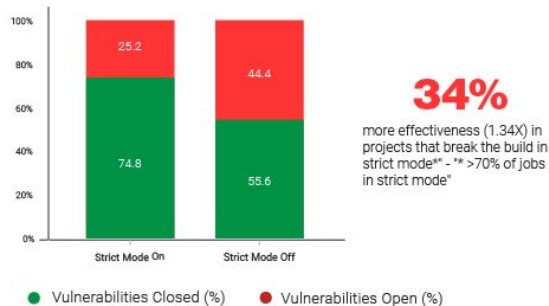
By asking to **break the build**, we are encouraging organizations to implement an aid to improve decision-making. We recommend you delegate the decision of whether something is released to production to an automated control that effectively checks that vulnerabilities are no longer present. This implies that your organization understands the risks of falling into a present-biased mindset by overlooking defects, and acts upon it. Companies are **2.45 times more effective** in fixing vulnerabilities by following this advice.⁹ Forcing the fix of all vulnerabilities before releasing imposes a cost now, but it will avoid greater costs in the future.

⁹ Fluid Attacks internal data.

How does vulnerability remediation behave when Fluid Attack's clients break the build?



How does remediation behave when Fluid Attacks' clients break the build in strict mode?



Break the build!

If you are continually assessing, in a committee or by yourself, whether a piece of software that has open bugs or security weaknesses is ready to be released, chances are you go for the immediate gratification. However, you can set up a control that helps you avoid that option by forcing your team to fix every single defect and every single weakness. In the words of Daniel Kahneman, organizations are factories of decisions¹⁰ and we are noisy decision-makers.¹¹ **By *breaking the build*, you are designing a better factory and reducing the noise.**







“Organizations are factories of decisions and we are noisy decision-makers.”

Daniel Kahneman
Nobel Prize in Economics, 2002

¹⁰ UBS, What Determines Human Decisions? - <https://www.ubs.com/daniel-kahneman>

¹¹ How to Overcome the High, Hidden Cost of Inconsistent Decision Making - <https://hbr.org/2016/10/noise>

Fluid Attacks' plans offer flexibility for your vulnerability management program

	  Machine Effective Automation	  Squad Beyond Automation	  One-Shot
Control of the whole remediation process	✓	✓	✓
Continuous vulnerability reporting	✓	✓	
Attack Surface Manager (GraphQL API)	✓	✓	✓
Automatic SAST, DAST and SCA	✓	✓	
CI/CD DevSecOps	✓	✓	
Low rates of false positives	✓	✓	✓
Low rates of false negatives		✓	✓
Higher severity and more types of vulnerabilities		✓	✓
Manual SAST, DAST and SCA (Pentesting)		✓	✓
Optimization of vulnerability search with AI		✓	
Unlimited reattack and search cycles		✓	
Possibility to talk to an expert via ASM		✓	



Web <http://fluidattacks.com/>
E-mail info@fluidattacks.com
Phone +1 (415) 404 2154
Location 95 3rd St, 2nd Floor
San Francisco, CA
94103

Legal clause

This document contains Fluid Attacks INC proprietary information. The reader may use this information for purposes of documentation only and may not disclose this document's contents to third parties for any reason without the expressed written consent of Fluid Attacks INC.

Copyright 2022 Fluid Attacks - All rights reserved