Adobe

WHITE PAPER

Adobe Security Testing Reports Overview

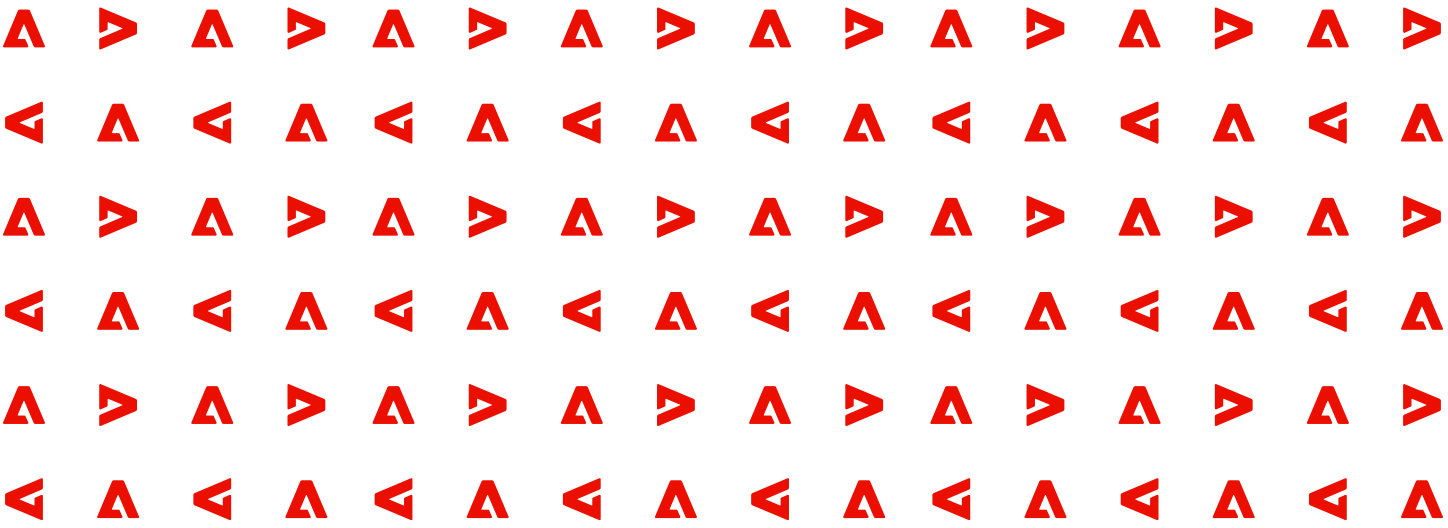


Table of Contents

Introduction

Ensuring the security and resilience of software and services has never been more important.

The number and types of potential threats grow every day, and the bad actors continue to become ever-more sophisticated — and powerful. What used to be the playground of amateur hackers is now a chessboard for extremely dangerous adversaries and nation-states. However, while the stakes have never been higher, product security testing has not evolved to keep up with these adversaries, leaving companies vulnerable to serious financial and reputational damage — or worse.

To combat the threats, most companies either test more frequently or conduct a greater number of tests. But neither of these approaches is sufficient to stop today's adversaries. There are a few reasons for this. First, no standardized metrics exist in the testing industry, so one pen-test vendor's "high" severity vulnerability could be another vendor's "medium." The result is customer confusion and a lack of an apples-to-apples method of comparison to determine the actual level of risk.

Second, outsourced penetration testing companies typically use widely accessible lists of "common exploits," such as the OWASP Top 10 or the SANS Top 25 checklists, to test product security. While this type of testing is an important starting point, solely focusing on these lists distracts from what really matters: exploitability.

Vulnerability vs. Exploitability

At this point, we need to define some terms:

- A *vulnerability* is a security flaw or weakness in system security procedures, design, implementation, or internal controls that hypothetically could be abused in order to cause harm to system resources or data.

- *Exploitability* means that a vulnerability has a proven and defined path to perform an attack along with a demonstrable security impact; It also considers the likelihood of abuse based on adversary signals and interest.

In other words, a vulnerability merely represents a theoretical possibility of attack success, whereas exploitability deals with reality—demonstrable, provable evidence of successful attacks. Rather than testing more frequently or running more tests, staying ahead of adversaries requires smarter testing: testing what they are interested in exploiting. To do this, product security testing must become more sophisticated and adversary-aware, focusing on the real-world exploitability of findings rather than theoretical vulnerabilities.

Focusing on demonstrably exploitable vulnerabilities also helps more effectively move the needle in overall product security by eliminating time-wasting "wild goose chases" and frustrating "noise" for product teams. Clear guidance for product teams about what really matters in improving security posture also helps improve the overall relationship between product and security teams.

What is Adversary-Aware Testing?

Adobe focuses on an objective measurement of the security posture of our products using an adversary-aware testing process.

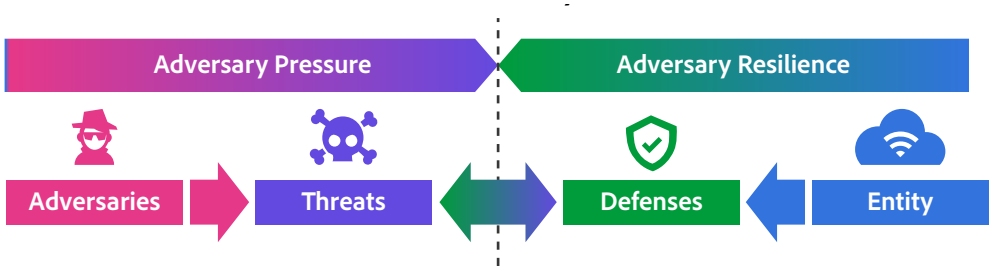


Figure 1: Adversary-Aware Testing

Each Adobe product is tested and measured against proven, exploitable threats. Upon conclusion of the testing, the results are published in a product-specific Adobe Security Testing Report (STR) on the Adobe Trust Center. These reports reflect all testing we conduct on a product, including our internal penetration testing, bug bounty reports, continuous automated testing scans, as well as third-party penetration tests. With these reports, customers gain both transparency into Adobe's testing processes and peace of mind that Adobe's products are as resilient as possible against adversary attacks.

The Adobe Security Testing Plan

Adobe's Security Testing Plan provides a comprehensive and repeatable method to help us more effectively zero-in on exploitable vulnerabilities. The plan is used to guide testing at all stages of the software development lifecycle from inception through deployment. For more detail, please see the Adobe Security Testing Lifecycle section below.

Currently, the Adobe Security Testing Plan defines six (6) security categories* on which we focus our product testing. These categories represent the most common avenues of attack exploited by adversaries and drive the direction provided to all Adobe product teams to help improve their security posture.

- **Keep your secrets safe** — Prevent exposure of digital authentication credentials, including passwords, keys, APIs, and tokens. For example, secrets may leak internally to employees or publicly, where anyone could have access if they are not properly secured. A malicious actor does not need to work too hard to access actual credentials if a developer eschews best practices in this area.

- **Mind your assets** — Be vigilant and transparent with the settings, configurations, and records of assets. For example, neglecting DNS records can result in subdomain takeover or point to a site that will serve malicious content to users.

- **Validate your inputs** — Restrict what code or command can be entered to only the properly formatted data. Adversaries often attempt to "trick" a product by inputting a code or a command that makes the system behave in a manner that wasn't intended. For example, uploading an incorrect file type or inputting improper text, code, or a command could allow an adversary to inject a malicious payload into the product.

- **Access matters** — Implement effective identity management and authentication controls to help ensure appropriate resource access by role or system. For example, inadequate access control measures can lead to unauthorized access, exposure of sensitive information, or execution of disallowed operations.

- **Patch your dependencies** — Keep products up to date with required security updates and conduct proper ongoing maintenance. Services and dependencies in the software supply chain that have reached EOL (end of life) provide a potential path of exposure for adversaries to execute an attack. For example, failing to remediate known exploitable attack paths provides a "welcome mat" for adversaries to conduct a wide range of nefarious activities.

- **Know your adversaries** — Understanding the specific motives and vantage points of various adversaries helps implement proper countermeasures. For example, adversaries often present themselves in the form a legitimate customer while looking for ways to steal information. Situational awareness about the types of adversaries that are interested in a product and their intent helps enable a proactive defense.

Adobe Security's testing specialists work with the product engineering organization to test their product in each category using the steps defined in the Adobe Security Testing Plan to uncover exploitable vulnerabilities and measure the effectiveness of our security controls from an adversary perspective.

Our Six-Step Testing Process

Because it is a closed-loop process, the Adobe Security Testing Plan enables test traceability and continuous improvement. For example, we can easily track how many tests were run in each category of testing, how many vulnerabilities were found, and which controls or procedures failed. Traceability helps us determine the root cause of a failure, providing intelligence that helps Adobe improve its company-wide security posture.

The first three (3) steps defined below occur before official product testing begins. These steps entail prescriptive security actions for developers to apply in their product. Next, the Adobe Security testing team runs a vast number of test cases to verify that the product team implemented the prescribed security controls and measure the effectiveness of these controls. Based on the gaps found during the testing process, the Adobe Security team provides guidance and support to the product team to remediate any discovered exploitable threats.

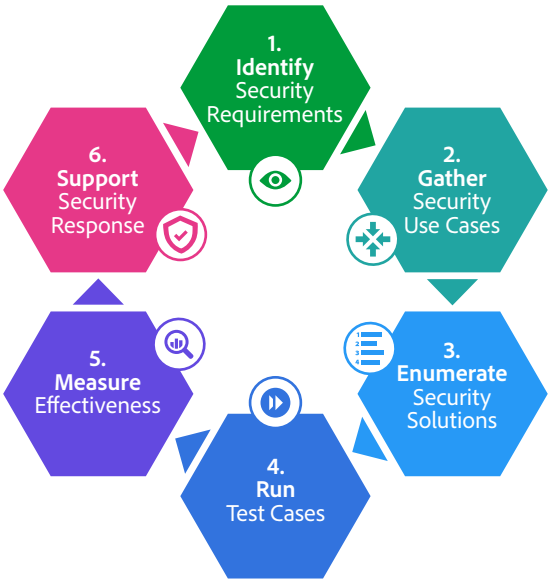


Figure 2: Adobe Security Testing Plan Process

1. Identify Security Requirements

The Adobe Security testing team defines a list of security requirements for each security category based on Adobe's documented security policies and standards. For example, to keep secrets safe, product teams must ensure they adhere to the Adobe-defined encryption standard.

2. Gather Use Cases

Next, the Adobe Security testing team defines various use cases for each security requirement. These use cases offer tailored guidance to product teams to help them

implement security requirements for specific functionality in the product. For example, an engineer working on developing an API would receive information from the Adobe Security team on known adversaries and their various attack methods.

3. Enumerate Security Solutions

For each security category, we proactively provide product teams with Adobe's recommended or preferred security solutions to address each security use case. In the example above, Adobe Security would provide the API developer with preferred solutions to thwart such attacks.

4. Run Test Cases

The Adobe Security organization runs test cases and integrated capabilities to test product workflows for each respective security use case. Test cases are based on known adversary attack vectors and our internal and external security research. During the testing process, the Adobe Security testing team partners with the product team to ensure results align with expectations and are measurable.

5. Measure Effectiveness

After the Adobe Security team runs the test cases to determine the resilience of the product against known adversary threats in each category and measures the following:

- Number of exploitable vulnerabilities discovered in each security category
- Total number of exploitable vulnerabilities found in the product tested

Using a simple percentage formula, where the total number of exploitable vulnerabilities equals 100%, we then determine the percentage distribution of the number of exploitable vulnerabilities in each category. The following example table lists the demonstrably exploitable findings, aligned to security categories, as defined in the Adobe Security Testing Plan:

Adobe Security Testing Plan Category	Percentage Distribution
Keep secrets safe	67%
Know your assets	0%
Validate your inputs	0%
Access matters	33%
Patch your dependencies	0%
Know your adversaries	0%
Others	0%

In the example table, 67% of the exploitable findings for this product occurred in the secrets category, while 33% were in the access category.

6. Support Security Response

The Adobe Security team then provides guidance and support to the product team to help them remediate the gaps found in the test cases to improve product resilience and security. After the product team implements our guidance and remediates the exploitable vulnerability, Adobe Security repeats the test cases to verify the exploitable vulnerability has been remediated. The step repeats until all known gaps are closed.

Adobe Security Testing Lifecycle

Adobe conducts both pre-deployment (also known as "shift-left") and post-deployment (also known as "shift-right") testing. Shift-left testing begins in the design process and ends with pre-deployment testing in our automated pipeline, while shift-right testing begins during the deployment process and continues until the end of life (EOL) of a release. Each testing cycle provides a chance to create new and revise existing test cases based on advances in security research and our adversary-aware testing, helping ensure that Adobe products are keeping up with the sophistication of adversaries.

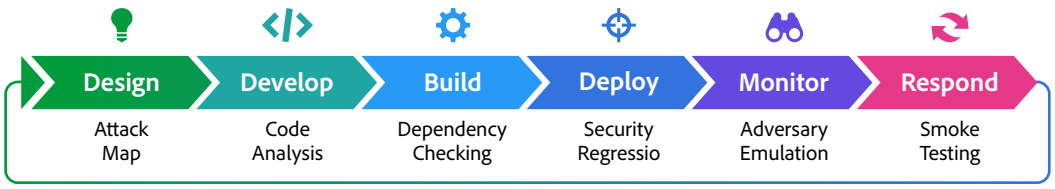


Figure 3: The Adobe Security Testing Lifecycle

Design

During the design phase, Adobe tests the proposed technical design for adherence to our corporate security policies, standards, and requirements as well as for resilience against potential attack patterns. We then build an attack map, also called a threat model, for the specific product, based on security industry information and adversary trends.

Adobe conducts this step during threat modeling, which includes a manual review of the security posture of the product architecture as well as automated security scanning, using the Adobe Security Testing Plan as a guide.

In addition, Adobe reviews all third-party vendors that store Adobe data in this stage using the Adobe Vendor Security Review (VSR) program to help ensure the secure handling, processing, and storage of Adobe data.

Develop

Adobe performs automated source code reviews using static code analysis during the development phase. We automatically scan every source-code pull request for security risks and flag it for remediation. Because these reviews are tightly integrated into Adobe's development workflow, we are able to minimize security risks in our products and services.

In addition to automated application testing, Adobe conducts manual mobile application testing and automation-assisted intellectual property (IP) code audit scans to improve the resilience of Adobe products and services.

Build

When a pull request is made during the build process, Adobe uses automated software component analysis to check for dependency vulnerabilities and flag them for remediation, when required.

Deploy

Adobe uses a variety of continuous automated testing techniques during the deployment phase including:

- **External Network Testing** — Performs penetration exercises from the perspective of an unauthenticated user attempting to gain privileged access to the infrastructure.
- **Internal Network Testing** — Tests the ability of an adversary who has accessed the internal network to discover and exploit vulnerable services from inside the network.
- **Application Testing** — Uses automated tools to conduct grey-box testing in combination with manual testing to determine both exploited and exploitable opportunities.

Monitor

Post-deployment, Adobe invests in adversary emulation by inviting external researchers and paid services to conduct verification testing. Through continuous bug bounties and our vulnerability disclosure program, Adobe recognizes that the security community is a vital participant in our goal to provide a safe and secure experience for customers and we welcome their contributions.

Additionally, we deploy holistic, company-wide red team campaigns to identify gaps in our defense-in-depth capabilities and improve our security processes and technologies. The Adobe red team uses adversary categories to improve and mature our company-wide resilience to adversaries.

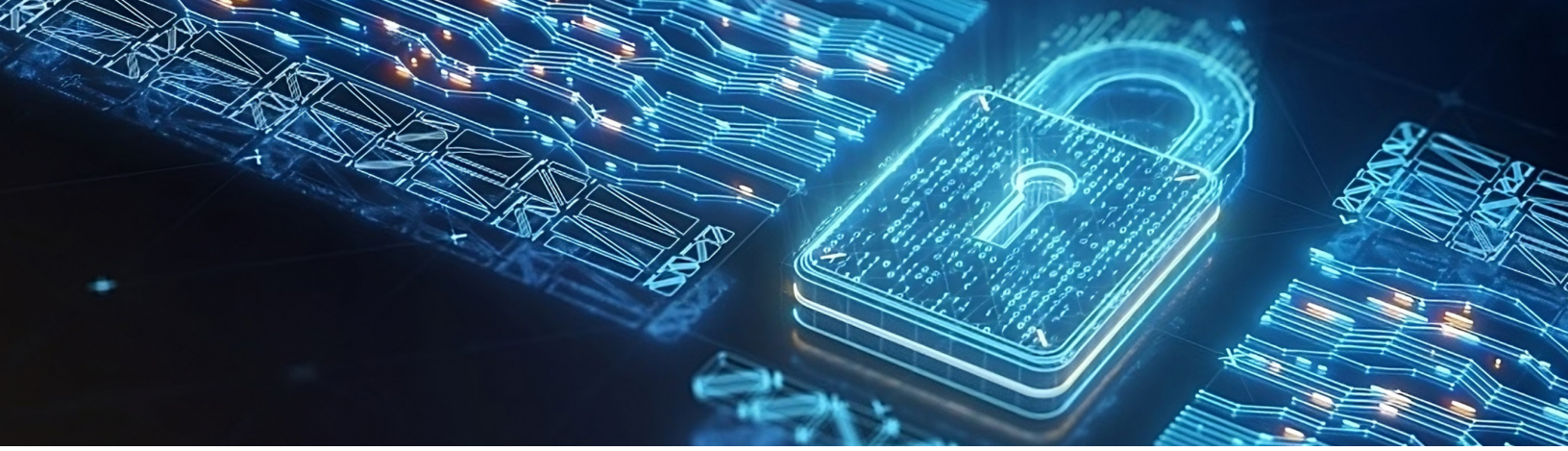
Externally, Adobe employs industry-leading vendors to perform annual, outsourced penetration tests of our application and network infrastructure to verify both test coverage and completeness. These tests are performed from the perspective of both an unauthenticated as well as an authenticated user, with the goal of bypassing user access control restrictions and/or gaining privileged access to the infrastructure through exploitation of application- and network-related vulnerabilities.

Respond

To ensure effective logging and detection of known and unknown threats, Adobe conducts smoke testing using its red team and internal researchers, helping ensure detection of specified adversary patterns and faster time-to-response in the event of a known adversary pattern.

Conclusion

Understanding adversary types and the parts of software they are most likely to exploit can be far more valuable than traditional static, list-based software testing processes. In practice, doing this requires a smarter approach to testing, one that is focused on the vulnerabilities that adversaries are demonstrably and provably interested in exploiting. Adobe seeks to raise bar in product security by focusing on a more objective measurement of the security posture of products, one that focuses on discovering exploitable vulnerabilities using an adversary- aware testing process. Through the Adobe Security Testing Reports, we give customers a transparent and standardized way to assess the security of our products based on calculable, objective risk as well as peace of mind that Adobe's products are as resilient as possible against adversary attacks.



Glossary

Term	Definition
Adobe Security Test Plan Process	Makes security test cases transparent and available for all engineering teams. These defined policies, standards, and solution, along with appropriate tests, provide clarity to achieve control adherence and resilience.
DevSecOps	Combines security with software development and IT operations in a set of practices that represent security as code.
Red team	Provides a real-world assessment of Adobe's security practices, controls, and response capabilities from an adversary perspective. Red team capabilities include continuous offensive security testing, exploit development, and systemic security issue discovery.
Security testing services	Enables DevOps (products and services teams) to detect and defend against real-world threats and test the security resilience of their products during the entire development lifecycle.
Shift-left testing	Consists of threat modeling, attack mapping, secure design and development, code reviews, application fuzzing, and dependency and build scans in the pre-deployment phases. Adobe Security's testing capabilities support the DevOps team help ensure application security in the early stages of development process and test the resilience of products during early development phases.
Shift-right testing	Focuses on validating the security of products and services after the development phase, including vulnerability scans, post-deployment testing, fuzzing, crawling, and exploit and abuse testing.
Smoke testing	Applies pressure to security defenses and implementations using scans, attack simulations, and other programmatic methods to determine if they are functioning to our expectations.

Adobe

11/23