Adobe Experience Cloud

Adobe

SECURABILITY REPORT

Adobe Target

Table of Contents

Overview

Protecting Adobe® and our customers' data is core to our business strategy; we are committed to safeguarding customer trust as demonstrated in our policies, standards, controls, and testing results. As part of this commitment, Adobe has created product-specific Adobe Securability Reports, which provide a consolidated and measured view of the security posture and results of adversary resilience testing for each product we offer.

The Adobe security organization provides comprehensive and continuous security testing that covers the entire software development process from inception through deployment. 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.

Methodology

The Adobe security organization implements a testing catalog that defines the type of testing required based on factors such as capability type, attack surface, risk, and adversary interest. Our testing methodology leverages both manual and automated tests, is designed to be as comprehensive as possible, and produces actionable results for product teams.

Adobe Open Test Plan Process

All security testing follows the Adobe Open Test Plan Process (OTPP), which helps enable test traceability and continuous improvement. This process identifies a list of security requirements per Adobe's security policies and standards and defines various use cases for each security requirement. Adobe-preferred security solutions are also recommended to address each security use case.

Our security organization defines the test cases and integrated capabilities to test product workflows for the respective security use cases. Our testing team also provides ongoing support to product teams throughout the testing lifecycle to ensure results align with expectations and are measurable. Through this process, we can assess and measure the effectiveness of our security controls from an adversary perspective, supporting Adobe's risk management objectives.

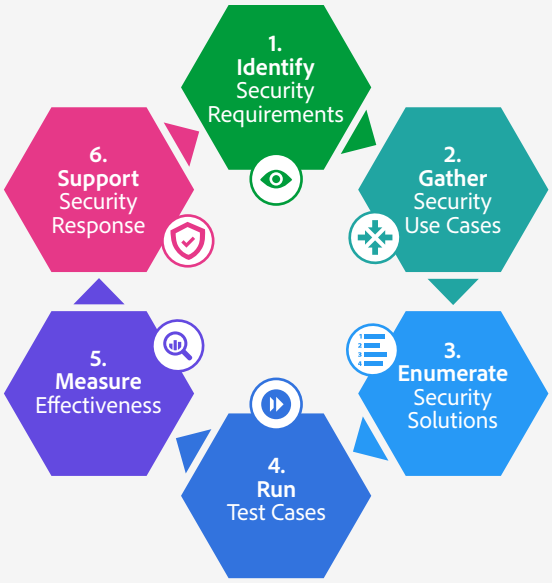


Figure 1: Adobe Open Test Plan Process (OTPP)

Adobe Security Testing Lifecycle

The Adobe security testing lifecycle includes eight (8) distinct steps, beginning early in the software development lifecycle. Using the Adobe OTPP, defined above, Adobe lists and catalogs the initial test cases that map to Adobe-defined policies, standards, and security requirements. Because this process supports versioning, our test suites continuously improve to meet evolving requirements and adversary trends.

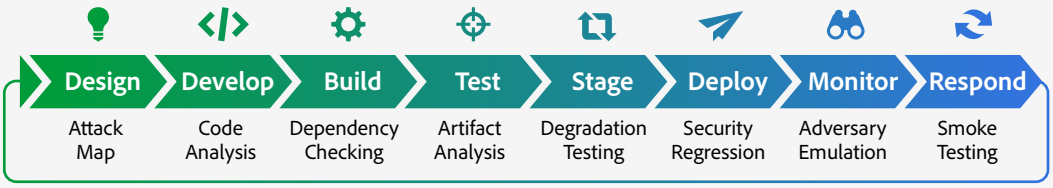


Figure 2: Adobe Security Testing Lifecycle

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 its regular product security reviews, which include a manual review of the security posture of the product architecture as well as automated security scanning, using the Adobe open test plan process 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.

Test

Adobe performs artifact analysis during development testing, using both manual and automated fuzzing and log analysis techniques.

Stage

Using a variety of vulnerability scans, Adobe conducts degradation testing against staging environments, including a full stack analysis of the environment.

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

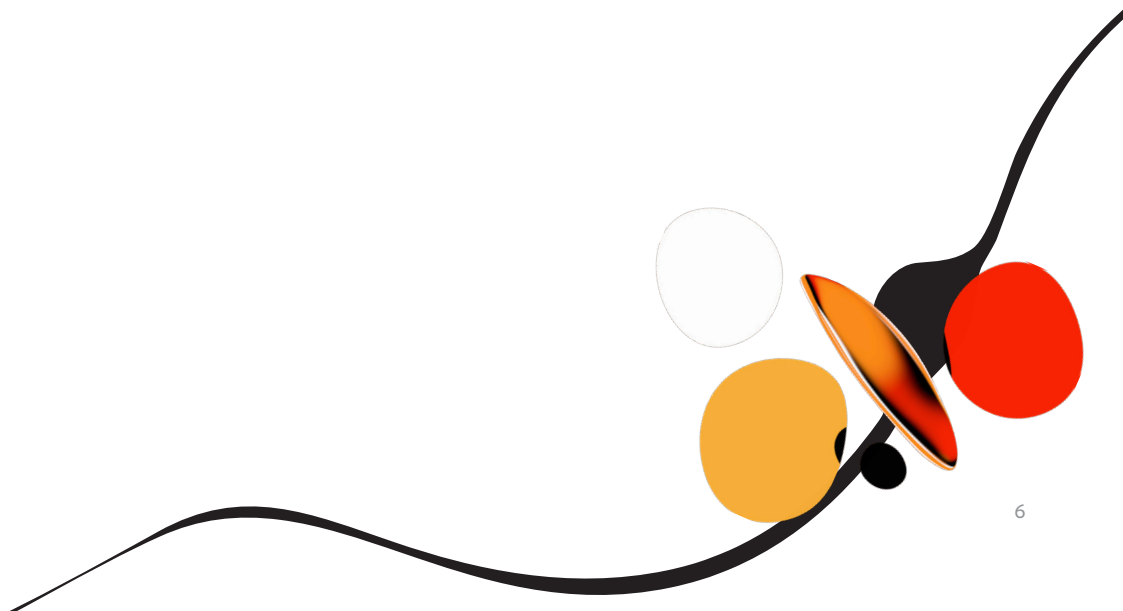
Post-deployment, Adobe invests in adversary emulation by inviting external researchers and paid services to conduct verification testing. Through continuous bug bounties and our vulnerability disclosure program, Adobe recognizes that the security community is a vital participant in our goal to provide 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 (attached to this report) 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. Adobe also runs continuous automated drift testing using exploit scenarios and vulnerability scans to ensure Adobe's security posture is maintained in production.

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.



Testing Scope

Adobe conducted a security assessment of Adobe Target using the Adobe security organization's testing services, including:

Scope Type	Components	Scope
Application and API Testing	Adobe Target services	[REDACTED].experiencecloud-stage.adobe.com https://experience-stage.adobe.com/?shell_ims=prod#/@[REDACTED]/target/
	UI components	exc-unifiedcontent.experience-stage.adobe.net

Figure 3: Adobe Testing Scope for Adobe Target

Test Results

Tests conducted by Adobe's security organization grouped discovered security vulnerabilities into the following two (2) categories:

- **Exploitable vulnerabilities:** Demonstrably exploitable or is concerning, with a publicly available exploit or proof of concept code in the wild; or
- **Informational findings:** Identified during testing to be not exploitable, typically attributed to security hygiene, best practices, or policies..

The following table lists the identified demonstrably exploitable vulnerabilities and corresponding remediation statuses:

#	Vulnerability	Status/Comment
1	Server-Side Request Forgery (SSRF) in Feeds functionality*	Resolved
2	Persistent cross-site scripting (XSS)*	Mitigated
3	Unauthorized write access	Resolved
4	Server path disclosure vulnerability	Resolved
5	CVE-2021-44228 (Apache Log4j RCE vulnerability exposure)	Resolved

Figure 4: Status of Exploitable Vulnerabilities

*Source: Outsourced pen test

The following table lists the demonstrably exploitable findings, aligned to security categories, as defined in the Adobe OTPP:

Open Test Plan/Security Category	Percentage Distribution
Keep secrets safe	0%
Know your assets	0%
Validate your inputs	80%
Access matters	20%
Patch your dependencies	0%
Know your adversaries	0%
Others	0%

Figure 5: Categories of Findings

Adobe stores any additional informational findings in our internal adversary intelligence data lake, a central location used in which Adobe stores testing and adversary data for analysis and gaining insights. These informational findings enable our teams to conduct in-depth analyses, improve hygiene, and stage input for our teams to conduct additional research.

Conclusion

Gaining and retaining the trust of our customers is one of Adobe's key values and is evident in our policies, standards, controls, and testing results. As part of our commitment to safeguard this trust and our customers' data, Adobe Securability Reports provide product-specific, comprehensive, and measured views of the security posture and results of adversary resilience testing for each Adobe product and service. For more information about Adobe Security, please see the Adobe Trust Center.

Glossary

Term	Definition
DevSecOps	A set of practices that combine security with software development and IT operations, through security as code.
Open Test Plan Process	An Adobe process that 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.
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.
Securability	A key performance indicator (KPI) that aids in measuring the effectiveness of critical security controls by testing for escapes from a known set of exploitable opportunities. Measuring securability provides a means of assessing the security control quality of an asset.
Security testing services	A portfolio of capabilities for DevOps (products and services teams) to leverage throughout their development lifecycle that help detect and defend against real-world threats and test the security resilience of their products.
Shift-left testing	Occurs pre-deployment and consists of threat modeling, attack mapping, secure design and development, code reviews, application fuzzing, and dependency and build scans. Adobe security testing capabilities that 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.

Certified by:

Digitally signed by
Anjul Bhambhri
Date: 2023-02-22
15:16:46-08:00

Senior Vice President

Appendix

Include Security, LLC

1 Hanson Place, #24A

Brooklyn, NY 11243

August 1, 2022

Adobe, Inc.

345 Park Avenue

San Jose, CA 95110-2704

To Whom It May Concern:

Include Security performed a security assessment of Adobe's Target Application. The objective of this assessment was to identify and confirm potential security vulnerabilities within targets in-scope of the SOW. The assessment team of 2 consultants completed an effort of 10 consulting days spanning from April 18th, 2022 - April 29th, 2022. The assessment was completed using a time box black box Assessment Methodology, outlined in the appendix "Statement of Coverage (Assessment Methodology)".

At the conclusion of the assessment, Include Security provided Adobe a report describing all findings in detail including recommendations for remediation, which included the following total findings per risk categorization:

Risk Categorization	Total per Category
Exploited Findings	
Critical	0
High	0
Medium	2
Findings Not Exploited	
Low	0
Informational	0
Total Categories of Findings	**2**

All the best,

Erik Cabetas

Managing Partner

Include Security, LLC

APPENDICES

Assessment Methodology

Once required information was gathered, IncludeSec performed the following activities:

- Network Security Assessment: IncludeSec performed an external vulnerability scan of in-scope FQDNs and provided an analysis of the results. The scan detected common network vulnerabilities and used a vulnerability identification strategy similar to that used by real world attackers.
  - IncludeSec worked to identify all results to remove any false positives and only manually confirmed findings were reported
  - Third party hosted servers were excluded from scans
- Application Abuse/Business Logic Testing: Using commercial tools, public tools, custom tools, and manual techniques IncludeSec worked to identify code patterns indicative of business logic flaws. In particular, the team sought to determine whether attackers could:
  - Obtain inappropriate access to other users' information
  - Obtain inappropriate access to sensitive and/or private information, such as billing information, account/app settings, etc.
  - Successfully make unauthorized changes
  - Successfully bypass business logic rules around account settings changes
  - Bypass authentication and authorization mechanisms
  - Bypass intended filtering actions
  - Escalate privileges and/or access site administration area from normal users
  - Hijack other users' accounts or sessions
  - Violate access controls placed by the site administrator
- Web Application Security Testing: Using commercial tools, public tools, custom tools, and manual techniques IncludeSec worked to identify code patterns indicative of implementation security bugs. In particular, the team sought to determine whether attackers could identify and exploit common and advanced web application security vulnerabilities, including:
  - Cross-Site Scripting (XSS)
  - Click-jacking
  - SQL, System Command, LDAP, XML, etc. injection points
  - Malicious content propagation (use the app as an attack proxy)
  - HTTP Redirects
  - Transport encryption verification (Non-SSL for sensitive pages)
  - Cipher Strength Analysis
  - Session management subversion
  - Session Fixation
  - Response Splitting
  - Cross-Site Request Forgery (CSRF)
  - Cookie Analysis