SYNOPSYS®

2021 Software Vulnerability Snapshot

An Analysis by Synopsys Application Security Testing Services

Table of contents

Overview

Synopsys Cybersecurity Research Center (CyRC) researchers recently examined anonymized data from commercial software systems and applications tested by Synopsys application security testing services during 2020. The following report includes data from 3,900 tests conducted on 2,600 targets (i.e., software or systems). Almost all of the tests (98%) were intrusive "opaque box" and "semi-opaque box" tests, including penetration tests, dynamic application security tests, and mobile application security analyses.

Industries represented in the study

Types of Security Testing

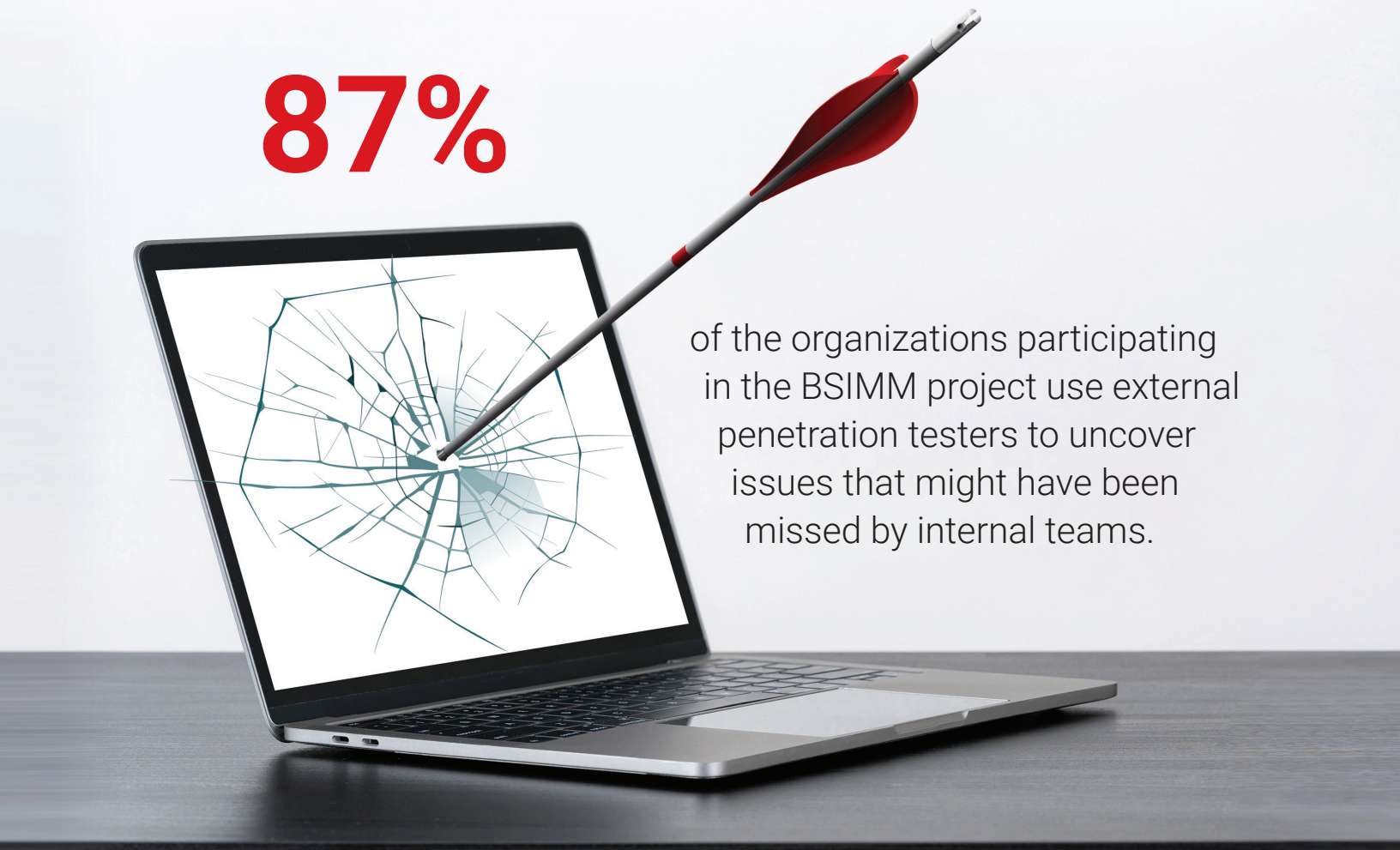
The majority of the tests—98%—were intrusive "opaque box" and "semi-opaque box" tests, including penetration testing, dynamic application security testing, and mobile application security analyses. Opaque box testing approaches the target's security state from an outsider's perspective, whereas semi-opaque box testing simulates an authenticated user with credentials—essentially extending opaque box testing with deeper insights. The tests were designed to probe running applications as a real-world attacker would, with the goal of identifying vulnerabilities that could then be triaged and remediated as necessary.

The targets tested were largely web (83%) and mobile (12%) applications, with the remainder either source code or network systems/applications. The industries represented included software and internet (29%), financial services (28%), business services (14%), manufacturing (10%), media and entertainment (9%), and healthcare (5%). The remaining 5% of test targets represented education, energy and utilities, and other verticals.

The tests focused on identifying software weaknesses, known as Common Weakness Enumerations (CWEs). These are flaws or errors in code that when left unaddressed can make software or a system vulnerable to attack.

About Third-Party Application Security Testing Services

Organizations use third-party application security testing services such as those offered by Synopsys for a variety of reasons. Some want to validate their own testing and ensure their internal security controls are working. Others need to comply with regulatory or business requirements that mandate a third-party assessment, while still others want to extend their software security testing without having to add specialized tools and staff. The 2021 BSIMM12 Insight and Trends report found that 87% of the organizations participating in the Building Security In Maturity Model (BSIMM) project use external penetration testers to uncover issues that might have been missed by internal testing.



Synopsys Application Security Testing Services 2020 by the Numbers

Number of Test Targets: **2,573**

Number of Tests: **3,937**

Tests That Uncovered Vulnerabilities:

3,828 **97%**

Number of Tests with High or Critical Severity Vulnerabilities:

1,420 **36%**

Total Number of Vulnerabilities Discovered:

28,501

Top Vulnerability Discovered: Missing Content-Security-Policy Header

52%

Top High-Risk Vulnerability Discovered: Stored Cross-Site Scripting

28%

Top Critical Vulnerability Discovered: SQL Injection

3%

Types of Tests

- Web App Pen Testing: **2,558 (67%)**
- Web App Dynamic Analysis: **630 (16%)**
- Mobile App Testing: **472 (12%)**
- Source Code Analysis: **88 (2%)**
- Network Security Pen Testing: **75 (2%)**

The Need for a Complete Software Security Testing Spectrum

Businesses that sell software, or sell products that include embedded software, can't afford software security, compliance, or quality issues compromising those products. Even businesses not directly engaged in selling software or software-driven products depend on software quality and security. For example, software drives the administrative systems for most payroll, billing, receivables, sales tracking, and customer records. Software controls production, manages inventories, directs warehousing, and runs the distribution systems that keep a business running.

Software is also the primary way that most businesses interact with and support customers, and these systems can also fall prey to attack. The giant credit risk assessment firm Equifax is not considered a software company, but a breach of the underlying software framework of an Equifax customer portal exposed the personal data of 143 million U.S. consumers in 2017.

Humans should perform the security tests they're the most effective at carrying out,

with their efforts augmented by automated testing.

Software risk *is* business risk, and to effectively manage the second, you must address the first. While "transparent box" testing such as static application security testing (SAST) can bring visibility to security issues early in the software development life cycle, SAST cannot uncover runtime security vulnerabilities. And some vulnerabilities cannot be easily detected by automated testing tools—they need human oversight to be uncovered.

For example, the only effective way to detect an insecure direct object reference (IDOR), an issue that allows attackers to manipulate references in order to gain access to unauthorized data, is by having a human perform a manual test. Clearly, there is no one best approach to application security testing. Humans need to perform the security tests they're the most effective at carrying out, with their efforts augmented by automated testing.

A full spectrum of application security testing is an essential component of managing software risk in today's world. When an organization lacks the needed human resources or tools to perform high-level opaque/semi-opaque box security testing such as penetration testing, or needs to vet its own software security controls, working with a third party such as Synopsys may be the best solution.

Vulnerabilities and Security Issues

Of the 3,900 tests, 97% uncovered some form of vulnerability in the targets. Thirty percent of the total were high-risk vulnerabilities, and 6% were critical-risk vulnerabilities. The easy availability of automated exploitation tools makes fixing high- and critical-risk vulnerabilities urgent whenever discovered.

High-risk vulnerabilities such as cross-site scripting (XSS) are issues that could allow attackers to access application resources and data. Twenty-eight percent of the test targets had exposure to reflected, stored, or DOM-based cross-site scripting vulnerabilities.

Critical-risk vulnerabilities allow attackers to execute code on a web application or application server and access sensitive data. Common critical-risk vulnerabilities include remote code execution and SQL injection—insertion of a SQL query via the input data from the client to the application. Three percent of the total test targets were vulnerable to some type of SQL injection.

The easy availability of automated exploitation tools makes fixing high- and critical-risk vulnerabilities urgent when discovered.

High-Risk Vulnerability	Percentage of Vulnerability in Total Test Targets
Reflected, Stored, or DOM-Based Cross-Site Scripting	28%
Improper Restriction of Excessive Authentication Attempts	6%
Vertical Privilege Escalation	6%
Missing Authentication	2%
HTTPS Not Enabled	2%

Table 1: Top High-Risk Vulnerabilities Found

Critical-Risk Vulnerability	Percentage of Vulnerability in Total Test Targets
SQL Injection	2%
Blind SQL Injection	1%

Table 2: Top Critical-Risk Vulnerabilities Found

Vulnerabilities Breakdown by the 2021 OWASP Top 10

The Open Web Application Security Project (better known as OWASP) Top 10 list represents a consensus among a large sampling of developers and web application security teams on the most critical security risks to web applications. In late 2021, the Top 10 list was updated for the first time since 2017, with three new categories added and others consolidated or with name and scope changes.

While intended by OWASP as an awareness document, many organizations use the list as a de facto application security standard. It's interesting to see how the tests conducted by Synopsys correlate with the OWASP Top 10.

Of the total 28,501 vulnerabilities discovered in the tests, 21,810—76%—fell into a 2021 OWASP Top 10 category. Table 3 lists the 10 most prevalent vulnerabilities Synopsys found, matched against five of the OWASP Top 10 categories. If the list had been extended further, all 10 of the OWASP categories would have been represented. For example, Vulnerable Third-Party Libraries in Use, which was found in 18% of the pen tests, correlates with the 2021 OWASP Top 10 category A06:2021—Vulnerable and Outdated Components.

Description	OWASP Top 10: 2021 Category	Percentage of Vulnerability in Total Vulnerabilities Found
Information Disclosure: Information Leakage	A01:2021—Broken Access Control	19%
Server Misconfiguration	A05:2021—Security Misconfiguration	18%
Insufficient Transport Layer Protection	A02:2021—Cryptographic Failures	8%
Authorization: Insufficient Authorization	A07:2021—Identification and Authentication Failures	7%
Application Privacy Tests	A07:2021—Identification and Authentication Failures	6%
Client-Side Attacks: Content Spoofing	A03:2021—Injection	5%
Fingerprinting	A07:2021—Identification and Authentication Failures	4%
Authentication: Insufficient Authentication	A07:2021—Identification and Authentication Failures	4%
Application Misconfiguration	A05:2021—Security Misconfiguration	3%
Client-Side Attacks: Cross-Site Scripting	A03:2021—Injection	2%

Table 3: Vulnerabilities Matched Against 2021 OWASP Top 10 Categories

A01:2021—Broken Access Control

Nineteen percent of the total vulnerabilities were related to the OWASP A01:2021—Broken Access Control category, which moved from the fifth to first listing in the 2021 OWASP Top 10. The OWASP team noted more occurrences of vulnerabilities that fit into this category in the applications tested than any other category. Notable CWEs included in this category are CWE-200: Exposure of Sensitive Information to an Unauthorized Actor, CWE-201: Exposure of Sensitive Information Through Sent Data, and CWE-352: Cross-Site Request Forgery.

Many of the vulnerabilities contained in the Broken Access Control category cannot be easily detected by automated testing tools.

It's worth noting that many of the vulnerabilities contained in the Broken Access Control category are more failures in business logic than actual vulnerability types, and they cannot be easily detected by automated testing tools. For example, IDOR issues, which allow attackers to manipulate references in order to gain access to unauthorized data, are included in this group. As mentioned previously, the only effective way to detect IDOR issues is by having a human perform a manual test.

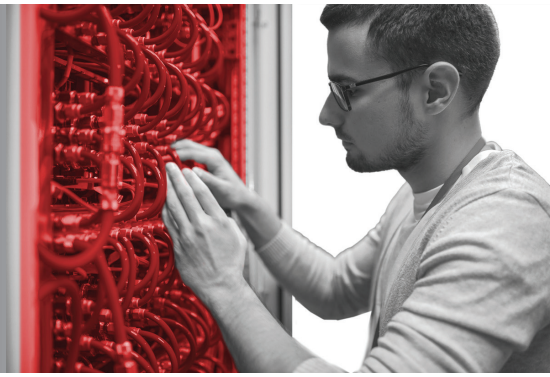
Of the total 28,501 vulnerabilities discovered in the tests, 21,810—76%—fell into an OWASP Top 10 category.

A02:2021—Cryptographic Failures

Eight percent of the total vulnerabilities uncovered in the tests related to the second category in the new OWASP Top 10, A02:2021—Cryptographic Failures, previously known as Sensitive Data Exposure.

This renamed category has moved up one position, from third to second, in the 2021 OWASP Top 10. The OWASP team describes the category as "more of a broad symptom than a root cause," as these vulnerabilities focus on failures related to cryptography, which can often lead to exposure of sensitive data. Notable CWEs included in this category are CWE-259: Use of Hard-Coded Password, CWE-327: Broken or Risky Crypto Algorithm, and CWE-331: Insufficient Entropy.

When combined in the OWASP A05:2021—Security Misconfiguration category, application and server misconfigurations represented 21% of the overall vulnerabilities.



A03:2021—Injection

The A03:2021—Injection category, now third in the 2021 OWASP Top 10, includes well-known vulnerabilities such as CWE-79: Cross-Site Scripting, CWE-89: SQL Injection, and CWE-73: External Control of File Name or Path. Seven percent of the total vulnerabilities found in the tests fell into this category.

A05:2021—Security Misconfiguration

When combined in the OWASP A05:2021—Security Misconfiguration category, application and server misconfigurations represented 21% of the overall vulnerabilities found in the tests. The OWASP team notes, "with more shifts into highly configurable software, it's not surprising to see this category move up" from sixth to fifth position in the OWASP Top 10. Notable CWEs included in this category are CWE-16 Configuration and CWE-611 Improper Restriction of XML External Entity Reference.

A07:2021—Identification and Authentication Failures

Previously known as Broken Authentication, the A07:2021—Identification and Authentication Failures category slid to seventh position in the OWASP Top 10 and now includes CWEs related to identification failures.

Twenty-one percent of the total vulnerabilities found in the tests belonged to this category, including what is identified in the tests as "fingerprinting," a security measure sometimes used to authenticate users. However, unless web servers are properly configured and monitored, fingerprinting can also provide attackers with valuable information such as OS type, OS version, SNMP information, domain names, network blocks, VPN points, and more.

Types of Tests

Sixty-seven percent of the tests performed were penetration tests—simulated attacks designed to evaluate the security of an application or system. Pen testing enables organizations to find and fix runtime vulnerabilities in the final development stages of software or after its deployment. Pen tests are often a compliance requirement of security standards. For example, the Payment Card Industry Data Security Standard (PCI DSS) requires penetration testing on a regular schedule or after any significant changes to the software or system.

Test Type	Percentage of Total Tests
Pen Testing	67%
Dynamic Analysis	16%
Mobile	12%
Static Analysis	2%
Network Security	2%
Other	0.1%

Table 4: Types of Tests Performed

Dynamic application security testing (DAST) and mobile application–specific tests comprised 16% and 12% of the total tests respectively. DAST is used to identify common coding weaknesses such as vulnerability to SQL injection, cross-site scripting, security misconfigurations, and other common issues detailed in the OWASP Top 10 and the CWE/SANS Top 25.

Synopsys DAST assessments include manual testing to uncover vulnerabilities that typically can't be found by out-of-the-box tools.

Synopsys DAST assessments include manual testing to uncover vulnerabilities that typically can't be found by out-of-the-box tools, such as some vulnerabilities pertaining to authentication and session management, access control, and information leakage.

Mobile application security testing (MAST) is used to uncover authentication and authorization issues, client-side trust issues, misconfigured security controls, cross-platform development framework issues, and vulnerabilities in application binaries running on the mobile device and corresponding server-side functionality.

Vulnerability	Number of Vulnerabilities	Percentage of Vulnerability in Total Pen Tests
Missing Content-Security-Policy Header	952	50%
Verbose Server Banner	861	45%
HTTP Strict Transport Security (HSTS) Not Implemented	736	39%
Weak SSL/TLS Configuration	725	38%
Cacheable HTTPS Content	633	33%
Reflected, Stored, or DOM-Based Cross-Site Scripting	531	28%
Weak Password Policy	500	26%
Insecure Content-Security-Policy Header	466	24%
Query String Parameter in HTTPS Request	457	24%
Clickjacking	408	21%
Excessive Session Timeout Duration	371	19%
TLSv1.0 Supported	355	19%
Unrestricted File Upload	347	18%
Vulnerable Third-Party Libraries in Use	336	18%
Verbose Error Messages (with Stack Trace)	331	17%

Table 5: Top Vulnerabilities Found in Pen Tests

Vulnerability	Number of Vulnerabilities	Percentage of Vulnerability in Total DAST Tests
Missing Content-Security-Policy Header	388	63%
HTTP Strict Transport Security (HSTS) Not Implemented	307	50%
Verbose Server Banner	305	49%
Cacheable HTTPS Content	242	39%
Excessive Session Timeout Duration	197	32%
Clickjacking	191	31%
TLSv1.0 Supported	179	29%
Weak SSL/TLS Configuration	162	26%
Reflected, Stored, or DOM-Based Cross-Site Scripting	162	26%
Insecure Content-Security-Policy Header	158	26%
Weak Password Policy	157	25%
Password Reset Username Enumeration	131	21%
Secure Cookie Attribute Not Set	130	21%
Unrestricted File Upload	127	21%
Unmasked NPI Data	123	20%

Table 6: Top Vulnerabilities Found in DAST Tests

Top Vulnerabilities Found in Mobile Tests Matched Against OWASP Risks

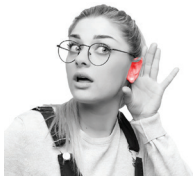
Vulnerability	OWASP Risk Category	Percentage of Vulnerability in Total Mobile Tests
Lack of Binary Obfuscation	M8: Code Tampering	30%
Application Allows Sensitive Data to Be Copied	M2: Insecure Data Storage	24%
Application Screenshot Information Disclosure	M4: Insecure Authentication	21%
Insecure configuration of Application Transport Security (iOS)	M3: Insecure Communication	20%
Lack of Certificate Pinning	M3: Insecure Communication	19%
Sensitive Data Stored Unencrypted in Local Storage	M4: Insecure Authentication	18%
No Jailbreak Detection	M8: Code Tampering	16%
Sensitive Data Logged to System Logs	A09:2021—Security Logging and Monitoring Failures	15%
Weak SSL/TLS Configuration	M3: Insecure Communication	14%
Jailbreak Detection Bypass	M8: Code Tampering	14%
Verbose Server Banner	A05:2021—Security Misconfiguration	14%
No Root Detection	M8: Code Tampering	13%

Table 7: Top Vulnerabilities Found in Mobile Tests Matched Against OWASP Risks

Although not as well-publicized as its overall Top 10 list, OWASP also publishes a mobile risks list, last updated in 2016. Table 7 matches the mobile test findings against the OWASP Top 10 Mobile Risks list of 2016. Two of the vulnerability categories listed are server-side issues, and they are shown with the comparable 2021 OWASP Top 10 list categories.

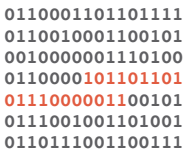
Insecure Data Storage

Twenty-four percent of the discovered vulnerabilities in the mobile tests were related to OWASP M2: Insecure Data Storage. These vulnerabilities could allow an attacker to gain access to a mobile device either physically (i.e., accessing a stolen device) or through malware.



Insecure Communications

Fifty-three percent of the mobile tests uncovered vulnerabilities associated with insecure communications. General best practices include using certificates signed by a trusted provider and ensuring that application transport security is enabled for iOS devices.



Code Tampering

A lack of binary protection can result in a mobile app that can be quickly analyzed, reverse-engineered, and modified by an adversary. This vulnerability was found in 30% of the mobile tests. Other vulnerabilities falling into this category include No Jailbreak Detection, Jailbreak Detection Bypass, and No Root Detection.

Server Security Misconfigurations and Security Logging and Monitoring Failures

Verbose server banners—found in 14% of the mobile tests—provide information such as server name, type, and version number; this information could allow attackers to perform targeted attacks on specific technology stacks. The Security Logging and Monitoring Failures category encompasses insufficient logging, detection, and monitoring. Issues related to this category were found in 15% of the tests.

Conclusion and Key Takeaways

Vulnerability	Number of Vulnerabilities	Percentage of Vulnerability in Total Test Targets
Missing Content-Security-Policy Header	1,347	52%
Verbose Server Banner	1,263	49%
HTTP Strict Transport Security (HSTS) Not Implemented	1,108	43%
Weak SSL/TLS Configuration	1,002	39%
Cacheable HTTPS Content	918	36%
Reflected, Stored, or DOM-Based Cross-Site Scripting	723	28%
Weak Password Policy	717	28%
Insecure Content-Security-Policy Header	632	25%
Query String Parameter in HTTPS Request	610	24%
Clickjacking	608	24%

Table 8: Top 10 Vulnerabilities Found

Testing with a Full Spectrum of Security Tools

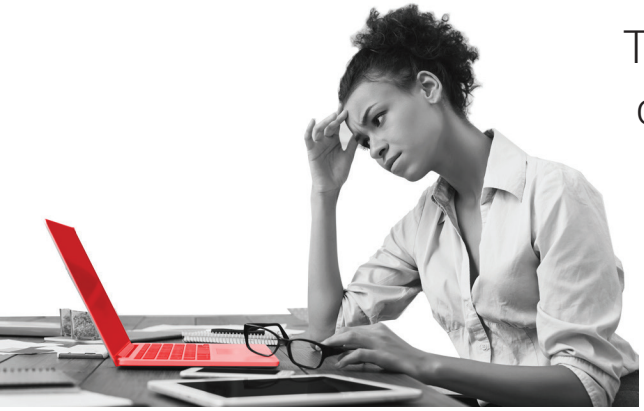
As noted earlier in this report, many organizations use third-party application security testing services to validate their own testing, ensure that their internal security controls are working, comply with regulatory or business requirements that mandate a third-party assessment, or extend their own software security activities without having to add more staff or new tools.

The findings detailed in Table 8 indicate that the majority of development teams had probably conducted their own transparent box security testing—such as static analysis—earlier in the software development life cycle and addressed many vulnerabilities before having Synopsys examine the running applications/systems with dynamic analysis, mobile, and penetration tests. It's also clear from the results that the best approach to security testing is to use a spectrum of the tools available to help ensure an application or system is secure.

For example, 28% of total test targets had some exposure to a cross-site scripting attack, one of the most prevalent and destructive high-/critical-risk vulnerabilities impacting web applications. While SAST testing can detect many common vulnerabilities, it is limited to discovering vulnerabilities that occur in the code itself. Many XSS vulnerabilities occur only when the application is running.

Even Lower-Risk Vulnerabilities Can Be Exploited to Facilitate Attacks

Sixty-four percent of the vulnerabilities the tests discovered are considered minimal-, low-, or medium-risk. That is, the issues found are not directly exploitable by attackers to gain access to systems or sensitive data. Nonetheless, surfacing these vulnerabilities is not an empty exercise, as even lower-risk vulnerabilities can be exploited to facilitate attacks. For example, verbose server banners—found in 49% of the tests—provide information such as server name, type, and version number that could allow attackers to perform targeted attacks on specific technology stacks.



The prevalence of cross-site scripting, clickjacking, and cross-site leak exploits makes a strong argument for having a secure content security policy to protect against various types of attacks, especially cross-site scripting.

Implementing or securing protections such as a content security policy (CSP) can provide an added layer of security that helps detect and mitigate certain types of attacks, including cross-site scripting and data injection attacks. An insecure or absent CSP—which was found (or, more accurately, not found) in 77% of the tests—may be considered a low-risk concern. However, the prevalence of cross-site scripting, clickjacking, and cross-site leak exploits makes a strong argument for having a secure CSP as an effective second layer of protection against various types of attacks, especially cross-site scripting.

An Urgent Need for a Software Bill of Materials

As shown in Tables 5 and 6, the pen and dynamic analysis tests found essentially the same types of vulnerabilities, with some variation in percentages found. However, of note is the number of Vulnerable Third-Party Libraries in Use, found in 18% of the pen tests. Although not detailed in this report, the same vulnerability was found in 33% of the static analysis tests Synopsys application testing services conducted in 2020.

Unrestricted File Upload	347	18%
Vulnerable Third-Party Libraries in Use	336	18%
Verbose Error Messages (with Stack Trace)	331	17%

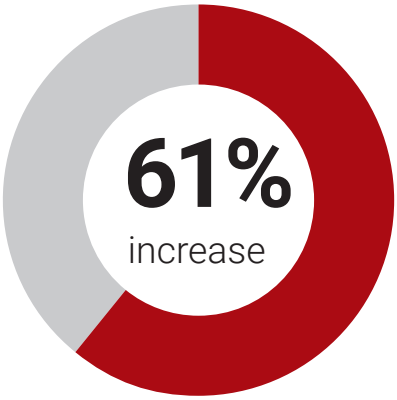
Highlight from Table 5: Top Vulnerabilities Found in Pen Tests

The vulnerability description correlates with the 2021 OWASP Top 10 category A06:2021—Vulnerable and Outdated Components. As OWASP notes, your software is likely vulnerable if

- You do not know the versions of all components you use (both client-side and server-side). This includes components you directly use as well as nested dependencies.
- The code being used is unsupported or out-of-date. This includes the OS, web/application server, database management system, applications, APIs, and all components, runtime environments, and libraries.

As the old security saw goes, "you can't fix problems you don't know you have." Most organizations typically use a mix of custom-built code, commercial off-the-shelf code, and open source components to create the software they sell or use internally. Often those organizations have informal—or no—inventories detailing exactly what components their software is using, as well as those components' licenses, versions, and patch status. With many companies having hundreds of applications or software systems in use, each themselves likely having hundreds to thousands of different third-party and open source components, an accurate, up-to-date software Bill of Materials (SBOM) is urgently needed to effectively track those components.

The concept of SBOM comes from manufacturing, where the classic BOM is an inventory detailing the items included in a product. When a defective part is discovered, the manufacturer knows precisely what product is affected and can begin the process of repair or replacement.



BSIMM12 data indicates a 61% increase in the "identify open source" activity over the past two years, due to the prevalence of open source components in modern software.

Similarly, more and more organizations are working to maintain an accurate, up-to-date SBOM that includes an inventory of third-party and open source components to ensure their code is high-quality, compliant, and secure. For example, BSIMM12 data indicates a 61% increase in the "identify open source" activity over the past two years, due to the prevalence of open source components in modern software and the rise of attacks using popular open projects as vectors.

And the demand for SBOMs continues to grow. In its 2020 Magic Quadrant for Application Security Testing, Gartner predicted, "By 2024, the provision of a detailed, regularly updated software Bill of Materials by software vendors will be a non-negotiable requirement for at least half of enterprise software buyers, up from less than 5% in 2019."

And while the 2021 executive order on Improving the Nation's Cybersecurity is primarily directed at federal departments, agencies, and contractors, its requirement for third parties to provide government software purchasers with a comprehensive SBOM will likely have a broad impact across critical infrastructure sectors and related technology suppliers.

About CyRC Research

The mission of the Synopsys Cybersecurity Research Center (CyRC) is to publish security research that helps organizations better develop and consume secure, high-quality software. Our recent security and software quality reports include the "Open Source Security and Risk Analysis" (OSSRA) report, and "Peril in a Pandemic: The State of Mobile Application Security."

SYNOPSYS®
CyRC

SYNOPSYS®

The Synopsys difference

Synopsys provides integrated solutions that transform the way you build and deliver software, accelerating innovation while addressing business risk. Our unmatched expertise helps you plan and execute any security initiative. We offer the most comprehensive product portfolio in the market, and it interoperates with third-party and open source tools. This open, pragmatic approach empowers your organization to leverage existing investments in testing tools to build the security program that best meets your needs. With Synopsys, your developers can secure code as fast as they write it. Your development and DevSecOps teams can automate testing within development pipelines without compromising velocity. And your security teams can proactively manage risk and focus remediation efforts on what matters most to your organization. Only Synopsys offers everything you need to build trust in your software.

For more information, go to www.synopsys.com/software.

Synopsys, Inc.
690 E Middlefield Road
Mountain View, CA 94043 USA

Contact us:
U.S. Sales: 800.873.8193
International Sales: +1 415.321.5237
Email: sig-info@synopsys.com