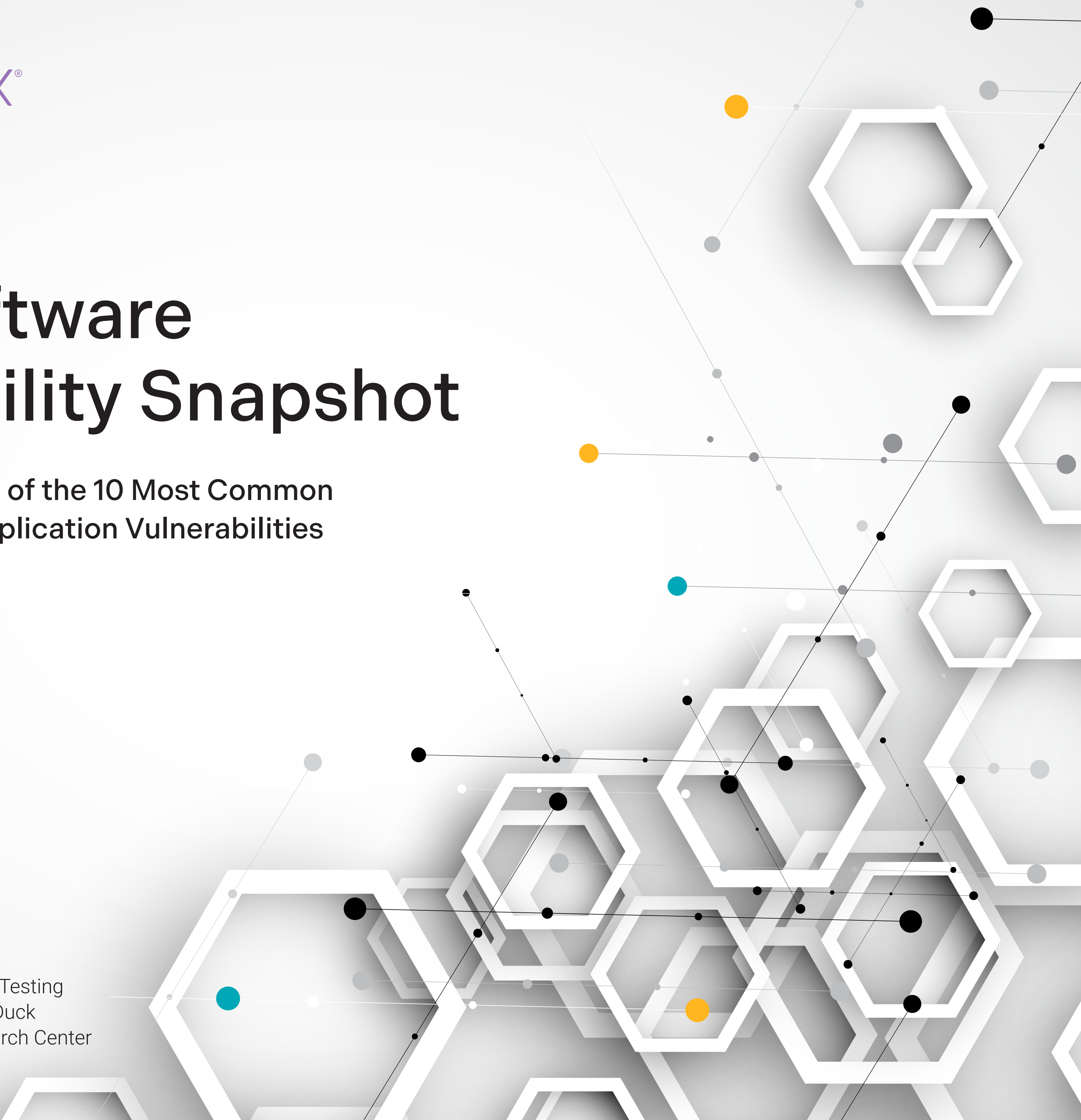
2023 Software Vulnerability Snapshot

A Three-Year Analysis of the 10 Most Common Web and Software Application Vulnerabilities



Black Duck Security Testing Services and Black Duck Cybersecurity Research Center

Table of contents

Overview

To produce the "Software Vulnerability Snapshot" report, Black Duck Cybersecurity Research Center researchers and Black Duck Security Testing Services consultants used anonymized data from three years of tests conducted on commercial software systems and applications.

The Black Duck tests shed light on persistent vulnerabilities that are significant challenges to web and software application security, especially the top vulnerabilities related to

- Information disclosure/leakage and privacy
- Misconfigurations
- Insufficient transport layer protection

The tests underscore the ongoing dangers posed by vulnerable third-party libraries and the need for robust software supply chain security in software development environments, where well over 90% of software contains open source. The following data also shows the value of extending application security testing beyond basic static analysis.

If you're in charge of a software security program, getting a deeper view into software risk can help you plan strategic improvements in your security efforts. If you're looking at security from the tactical side, you can use the information in this report to present a business case for the need to expand your security testing with third-party assistance.

Security Issues Found in the Tests

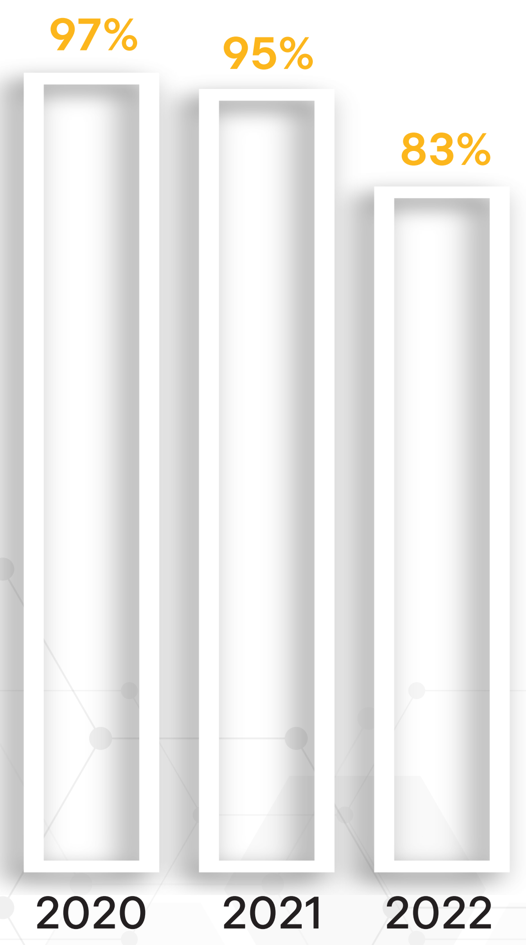
In 2020, 97% of the tests found vulnerabilities. In 2021, that figure dropped to 95%, and it decreased even further in 2022, to 83%. Over all three years of collected data, 92% of the tests uncovered vulnerabilities in the target applications.

The continual decrease in overall vulnerabilities is an encouraging sign that development teams are improving their ability to write error-free code and that practices such as code reviews, automated testing, and continuous integration are helping to reduce common programming errors.

Advancements in programming languages and integrated development environments (IDEs) now provide built-in checks and tools that help developers catch errors before they become significant issues. In the case of popular open source projects, many communities have also ramped up their scrutiny of code, leading to higher quality standards.

Unfortunately, the same can't be said for less popular or older open source projects. According to some reports, nearly 20% of open source projects across Java and JavaScript that were maintained in 2022 are no longer being maintained today, opening those projects to vulnerabilities and exploits.

Percentages of tests that found vulnerabilities



High- and Critical-Severity Vulnerabilities

Over all three years, 27% of the tests found high-severity vulnerabilities, and 6.2% revealed critical-severity vulnerabilities. Vulnerabilities allowing cross-site scripting (XSS) have consistently been one of the top high-severity vulnerabilities found in Black Duck tests over the years. Similarly, the top critical-severity vulnerability, SQL injection, has remained the leader from 2020 to 2022.

Breaking it down by year, we saw a small increase in high-severity vulnerabilities between 2022 and 2021 (25% vs. 20%) and a decrease (25% vs. 30%) when comparing 2022 to 2020. Critical-severity vulnerabilities were higher (6.7%) in the 2022 tests than in 2021 (4.5%) or 2020 (6.1%).

Many medium- through critical-severity vulnerabilities require multilayered testing in order to be uncovered.

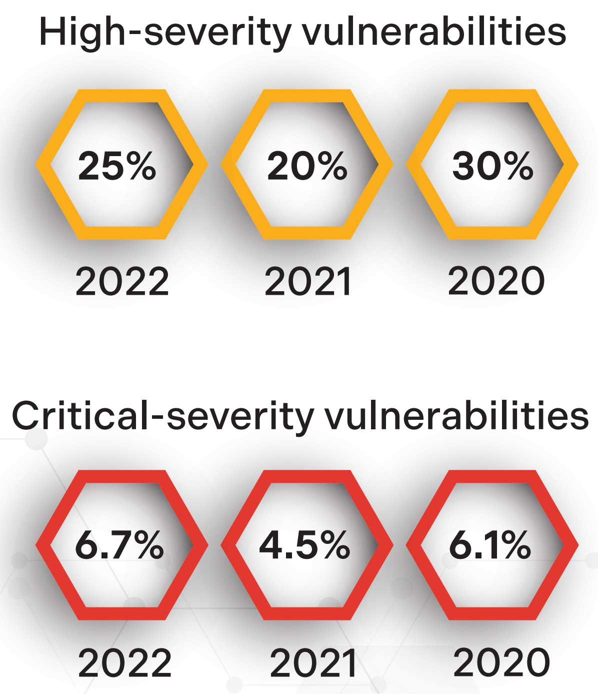
The numbers reflect that high- and critical-severity vulnerabilities have been increasing in the past few years and hit an all-time high in 2022. Some 80% of CVEs reported in 2022 were either medium or high severity, with 16% deemed critical. While development teams are reducing overall vulnerabilities, the data indicates that many medium- through critical-severity vulnerabilities require more robust testing—such as penetration testing—in order to be uncovered.

	2022	2021	2020
Critical	28%	54%	77%
High	38%	46%	63%
Medium	60%	64%	72%

Figure 1. Reduction in percentage of vulnerabilities found in retests compared to original tests

Figure 1 illustrates that dynamic testing—such as some types of penetration testing and dynamic application security testing (DAST)—is an effective supplement to static application security testing (SAST). A sampling of retests (that is, quick validations that specific fixes have been applied) show customers reducing critical-, high-, and medium-severity vulnerabilities in each of the three years once vulnerabilities have been found. For example, in 2022, medium-severity vulnerabilities were found to have been reduced by 60%, while high- and critical-severity vulnerabilities were reduced by 38% and 28% respectively.

Percentage of high- and critical-severity vulnerabilities 2020-22



Defining Severity Levels

Severity levels reflect the risk a vulnerability poses to network security, as rated on the CVSS v3 scale. Critical-severity vulnerabilities have a CVSS score ranging from 9.0 to 10.0 and an exploit or proof-of-concept code publicly available or being actively exploited. They include, for example, command, code, and SQL injection exploits.

High-severity vulnerabilities have a CVSS score ranging from 7.0 to 8.9 and are usually more difficult to exploit than critical vulnerabilities. They still should be reviewed and addressed promptly, considering factors such as the business criticality of the at-risk application/system and the threat landscape.

With more attackers using automated exploitation tools that can attack thousands of systems in a matter of seconds, fixing high- and critical-risk vulnerabilities can become urgent whenever those vulnerabilities are discovered, not least because well over half of reported vulnerabilities are exploited within a week of disclosure.

Security or vulnerability issues in deployed applications tend to cascade downhill, not only through their potential of disrupting an organization's (or its customers') business operations, but also through their impact on the entire SDLC, and in extension, the software supply chain.

In fact, over 80% of respondents in the Black Duck "Global State of DevSecOps 2023" report said that the need to address a critical security/vulnerability issue had impacted their organization's software delivery schedule during 2022-23.

About Black Duck Security Testing Services

Development teams are being asked to build increasingly complex software faster than ever before, but skilled, trained resources are at a premium.

While belonging to different industries and organizations, all Black Duck Security Testing Services customers share a common need—to supplement their testing coverage. Development teams are being asked to build increasingly complex software faster than ever before, but skilled, trained resources are at a premium, especially when it comes to software security.

Industries represented in the study include

31%	Software and internet, systems integration and related services, computers and electronics
29%	Financial services, insurance
12%	Business services
8%	Manufacturing, transportation and storage, wholesale and distribution
4%	Healthcare
6%	Other*

*Includes members of the travel and leisure, education, and utilities industries as well as the public sector.

Three-year averages (2020–22) of industries represented in this report

A Black Duck report on the strategies, tools, and practices impacting software security notes that over 33% of 1,000 respondents pointed to inadequate security training as a major roadblock, a figure closely followed by a shortage of security personnel (31%). Thirty-three percent of the report's respondents also said that external consultants were aiding in security testing for their organizations.

It can be invaluable to contract third-party security testers to gain an unbiased view of an organization's security posture. In fact, the "Building Security in Maturity Model" (BSIMM) report has found that more than 88% of the organizations participating in the BSIMM project use external penetration testers to supplement their security activities. These tests can uncover issues that might have been missed by internal testing, potentially highlighting a weak link in an organization's security practices.

Even if you feel your organization has adequate security testing coverage, you may want to validate your own testing and ensure that your internal security controls are working. Or you may need to comply with regulatory, customer, or other requirements that mandate third-party assessments. For example, Requirement 11 of PCI DSS 4.0 specifically mandates the performance of regular penetration testing. This requirement is applicable to merchants that need to do a formal audit and is also applicable to all service providers.

The Health Insurance Portability and Accountability Act (HIPAA) requires healthcare practitioners to protect electronically stored, health information using technical safeguards to ensure the confidentiality and security of the information. While HIPAA does not specifically require a penetration test or vulnerability scan, it does mandate a risk analysis, which effectively requires covered entities to test their security controls.

The section of HIPAA that addresses "evaluation" specifically calls out "periodic technical and nontechnical evaluation" methods. And in its HIPAA guidance, the National Institute of Standards and Technology (NIST) cites external and/or internal penetration testing as the recommended method of meeting these technical evaluation requirements, where reasonable and appropriate.

In the financial services world, the Financial Industry Regulatory Authority (FINRA) establishes the cybersecurity rules for financial organizations and recommends running penetration tests regularly and after key events, such as significant changes to a firm's infrastructure or access controls. Since 2022, and with an enforcement deadline in June 2023, the Gramm-Leach-Bliley Act (GLBA) explicitly requires financial institutions to perform annual penetration testing and vulnerability scanning as part of their security activities.

A Black Duck report found that

over 33%

of 1,000 respondents pointed to inadequate security training as a major roadblock

over 31%

cited a shortage of security personnel as a problem

Black Duck Security Tests in Detail

The Black Duck tests probe running applications as a real-world attacker would, incorporating both "black box" and "gray box" testing, with the goal of identifying vulnerabilities that can then be triaged and remediated as necessary. Black box testing approaches the target's security state from an outsider's perspective, whereas gray box testing simulates an authenticated user with credentials—essentially extending black box testing with deeper insights. Tests were largely conducted on web (82%) and mobile (13%) systems/applications, with a smaller number of network (3.0%) and source code (2.0%) targets.

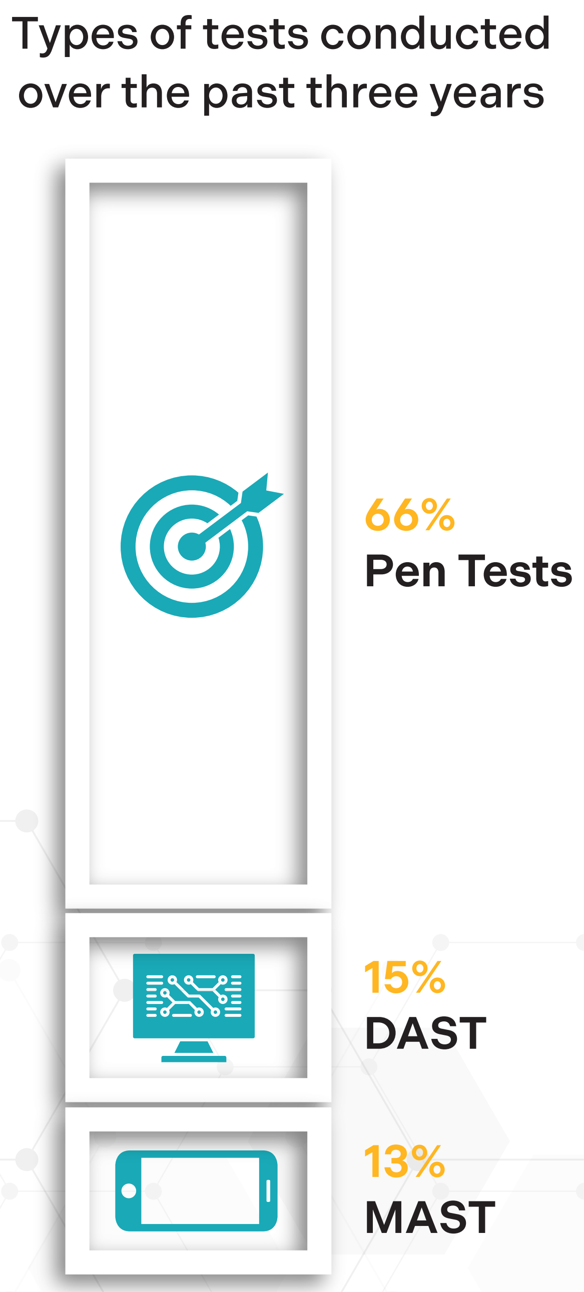
Pen Testing

Sixty-six percent of the tests were penetration tests—an authorized simulated attack performed on a computer system to evaluate its security. Penetration testers use the same tools, techniques, and processes as attackers to find and demonstrate the business impacts of weaknesses in a system. Penetration tests examine whether a system is robust enough to withstand attacks from authenticated and unauthenticated positions, as well as a range of system roles.

Often required to comply with industry regulations and standards, external pen testing brings added benefits such as an unbiased viewpoint of your security posture, as well as an accurate simulation of potential threats and vulnerabilities that external adversaries might exploit.

A comprehensive approach to pen testing is essential for optimal risk management. Black Duck pen testing may include targets such as

- Web apps. Testers examine the effectiveness of security controls and look for hidden vulnerabilities, attack patterns, and any other potential security gaps that can lead to a compromise of a web app.
  - Mobile apps/devices. Using both automated and extended manual testing, testers look for vulnerabilities in application binaries running on a mobile device and the corresponding server-side functionality. Server-side vulnerabilities include session management, cryptographic issues, authentication and authorization issues, and other common web service vulnerabilities. Vulnerabilities in application binaries can include authentication and authorization issues, client-side trust issues, misconfigured security controls, and cross-platform development framework issues.
  - Networks. This testing identifies critical security vulnerabilities in an external network and systems. Experts employ a checklist that includes test cases for encrypted transport protocols, SSL certificate scoping issues, use of administrative services, and more. Three percent (2.7%) of the pen tests conducted over all three years were network security penetration tests.



– APIs. Both automated and manual testing techniques are used to cover the OWASP API Security Top 10 list. Some of the security risks and vulnerabilities testers look for include broken object-level authorization, user authentication, excessive data exposure, lack of resources/rate limiting, and more.

Expert pen testers thinking and acting like adversaries introduce a needed human element, as they analyze data to target their attacks and test systems and websites in ways automated testing solutions following a scripted routine cannot. Plus, some vulnerabilities can't be easily detected by automated testing tools—they require 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 gain access to unauthorized data, is by performing a manual test.

DAST

Dynamic application security tests (DAST) were 15% of the tests conducted over all three years. DAST is a method of application security (AppSec) testing in which testers examine an application while it's running but have no knowledge of the application's internal interactions or designs at the system level, and no access or visibility into the source program. This type of black box testing looks at an application from the outside-in, examines its running state, and observes its responses to simulated attacks made by a testing tool. An application's responses to these simulations help determine whether the application is vulnerable and could be susceptible to a real malicious attack.

The goal is to find outcomes or results that were not expected and could therefore be used by attackers to compromise an application. Since DAST tools don't have internal information about the application or the source code, they attack just as an external hacker would—with the same limited knowledge and information about the application.

DAST solutions can find runtime vulnerabilities that are tough to spot through other security tests. These vulnerabilities include authentication and server configuration errors, code injection, SQL injection, and cross-site scripting errors.

As noted earlier in this report, human oversight is sometimes needed to develop a full software security picture. Black Duck DAST evaluations 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.

Human oversight is sometimes needed to develop a full software security picture.

MAST

Thirteen percent of the tests conducted over all three years were mobile application security testing (MAST) tests. Black Duck mobile application security testing focuses on analyses of mobile client-side code, server-side code, and third-party library analysis to systematically find and fix security vulnerabilities in mobile applications, without the need for source code. Black Duck uses a combination of proprietary static and dynamic analysis tools working together to discover vulnerabilities and offers multiple depths of analysis, tuning the level of testing according to the risk profile of each tested application.

Despite the widespread adoption of mobile apps, most development and security teams have been slow to embrace mobile security. Statistics from NowSecure's MobileRiskTracker indicate that 85% of mobile apps in public app stores include one or more high-risk vulnerability or violate one or more OWASP Mobile Application Security Verification Standards. And 70% leak private data, potentially breaching GDPR/CCPA and other privacy regulations.

85% of mobile apps in public app stores include one or more high-risk vulnerability.

Vulnerabilities Overview

The Open Web Application Security Project (OWASP) Top 10 and Common Weakness Enumeration (CWE) Top 25 are lists of the most common and dangerous security vulnerabilities. Both are based on data from a variety of sources, including security researchers, vulnerability databases, and incident reports.

The OWASP Top 10 list represents a consensus among a large group of developers and web application security teams on the most critical security risks to web applications. The CWE Top 25 list was created by MITRE in collaboration with the SANS Institute to rank the 25 most dangerous software vulnerabilities. While intended as awareness documents, many organizations use both lists as de facto security standards to measure the effectiveness of their security controls.

The OWASP Top 10 list focuses on security risks specific to web applications, whereas the CWE Top 25 list focuses on software weaknesses and their related CWEs.

As shown in Figure 2, there is overlap between the two lists, even when their terminology differs. For example, OWASP category A05:2021—Security Misconfiguration is related to various weaknesses listed in the CWE Top 25, such as CWE-798 (Use of Hard-Coded Credentials) and CWE-276 (Incorrect Default Permissions).

"Fingerprinting" (probing a web application to gain information) could arguably be placed in the OWASP Top 10 list's Security Logging and Monitoring Failures or Broken Access Control categories (we've listed it under the former), although it is not directly referred to in either category. Similarly, the closest CWE related to fingerprinting is probably Exposure of Sensitive Information to an Unauthorized Actor (CWE-200). As shown in rows 1, 4, 6, and 9 of Figure 2, information disclosure/leakage in various aspects was an issue found in many of the tests.

Top 10 Black Duck Vulnerability Categories 2022	Rank in 2021	Rank in 2020	Related OWASP Top 10/ Mobile Top 10 Category	Related CWE(s)
1. Information Disclosure: Information Leakage	1	1	A01:2021—Broken Access Control	Exposure of Sensitive Information to an Unauthorized Actor (CWE 200). Other access control management issues include missing/ incorrect authorizations (CWE-863) and vulnerability to client/server-side request forgery (CWE-918).
2. Server Misconfiguration	2	2	A05:2021—Security Misconfiguration	Security misconfigurations such as CWE-798 and CWE-276.
3. Insufficient Transport Layer Protection	3	3	A05:2021—Security Misconfiguration M3: Insufficient Transport Layer Protection	Security misconfigurations such as CWE-798 and CWE-276.
4. Insufficient Authorization	4	4	A01:2021—Broken Access Control M6: Insufficient Authorization	Exposure of Sensitive Information to an Unauthorized Actor (CWE 200). Other access control management issues include missing/ incorrect authorizations (CWE-863) and vulnerability to client/server-side request forgery (CWE-918).
5. Application Misconfiguration	9	9	A05:2021—Security Misconfiguration	Security misconfigurations such as CWE-798 and CWE-276.
6. Application Privacy Failures	5	5	A02:2021—Cryptographic Failures	Exposure of Sensitive Information to an Unauthorized Actor (CWE 200). Deserialization of Untrusted Data CWE-502.
7. Authentication: Insufficient Authentication	8	8	A07:2021—Identification and Authentication Failures	Improper or missing authentication such as CWE-287 or CWE-306.
8. Content Spoofing/Content Injection	6	6	A03:2021—Injection	SQL injection (CWE-89), command injection (CWE-78, CWE-77), code injection (CWE-94), cross-site scripting CWE-79.
9. Susceptibility to Fingerprinting	7	7	A09:2021—Security Logging and Monitoring Failures	Exposure of Sensitive Information to an Unauthorized Actor (CWE 200).
10. Exposure to Client-Side/ Cross-Site Scripting Attacks	Not in Top 10	10	A03:2021—Injection	Cross-site scripting CWE-79, cross-site request forgery (CWE-352).

Figure 2. Black Duck Top 10 vulnerability categories

Security Issues in Detail

As shown in Figure 2, the rankings of the vulnerability categories have remained largely unchanged over the years. An interesting exception is Application Misconfiguration, which moved to #5 in 2022 after two years at #9.

Misconfigurations can occur in applications, browsers, networks, operating systems, and servers. For example, Optus, the Australian telecommunications giant, experienced a severe breach in 2022 that exposed the personal data of 9.7 million customers. A misconfiguration in the company's firewall, allowing third-party contractors to access sensitive customer data, led to the breach.

Human error is typically the most common cause of misconfiguration issues. Organizations fail to adequately examine their applications and deployment scripts, leaving themselves vulnerable to attack. Configurations are altered during testing procedures and the alterations are not reverted to more secure settings. New hardware and software are not properly tested to meet organization's security requirements.

The Application Misconfiguration position jump illustrates the need for multiple security testing tools rather than relying on one to do the job. While organizations may be doing a good job lowering their overall number of coding vulnerabilities, they shouldn't rely on a single testing tool, especially when it comes to runtime environments. A broad spectrum of testing tools is needed to uncover issues, such as misconfigurations, in running applications.

The top 10 vulnerability categories found in the 2022 tests are the following:

1. **Information Disclosure**, also known as **Information Leakage**. This security issue occurs when sensitive information is exposed to unauthorized parties. For example, a website might leak data about users, such as usernames or financial information, through some type of security misconfiguration. An average 19% of the total vulnerabilities found over all three years of testing were directly related to information leakage issues.

   The OWASP team lists information disclosure under the A01:2021—Broken Access Control category and notes that more vulnerabilities fit into this category than any other category. Notable vulnerabilities in this category include exposure of sensitive information to an unauthorized actor, exposure of sensitive information through sent data, and cross-site request forgery.

An average 19% of the total vulnerabilities found over all three years of testing were directly related to information leakage issues.

2. **Server (Security) Misconfiguration**. A member of the OWASP A05:2021—Security Misconfiguration category, server misconfigurations represented an average 18% of the total vulnerabilities found over all three years of testing. While our results focus on server misconfigurations, security misconfigurations can happen at any level of an application stack, including the network services, platform, web server, application server, database, frameworks, custom code, and preinstalled virtual machines, containers, or storage (see category #5). Such flaws frequently give attackers unauthorized access to system data or functionality, occasionally even resulting in a complete system compromise.

An average 11% of vulnerabilities found over all three years of testing were related to insufficient transport layer protection.

3. **Insufficient Transport Layer Protection**. These are security weaknesses caused by applications not taking measures to protect network traffic. During authentication, applications may use SSL/TLS, but they often fail to use it elsewhere in the application, thereby leaving data and session IDs exposed.

   Many mobile applications have specific issues with insufficient transport layer security, to the point where OWASP has dedicated a category in its OWASP Mobile Top 10 list to it. As OWASP notes, “mobile applications frequently do not protect network traffic. They may use SSL/TLS during authentication but not elsewhere. This inconsistency leads to the risk of exposing data and session IDs to interception.” An average 11% of vulnerabilities found over all three years of testing were related to insufficient transport layer protection.

4. **Insufficient Authorization.** These vulnerabilities can potentially permit users to access data, content, or functionality that they should not be able to reach. This could happen when an application or system does not properly verify the identity of users or fails to enforce proper access controls.

   A mobile app transmitting a user’s roles or permissions unencrypted to a back-end system as part of a request is an example of insecure authorization. OWASP notes that the presence of IDOR vulnerabilities is often indicative of code not performing a valid authorization check. An average 9% of vulnerabilities found over all three years of testing were related to insufficient authorization issues.

Server misconfigurations represented an average 18% of the total vulnerabilities found over all three years of testing.

5. **Application Misconfiguration**. Another example of the prevalence of security misconfigurations, this is also the fifth-most-dangerous risk on the OWASP list of its top 10 vulnerabilities.

   Many applications come with developer features that are dangerously unsafe if not deactivated when deployed, such as debug and QA features. Configuration files that are not properly locked down may reveal clear text (that is, unencrypted text that can be read by anyone), and default settings in configuration files may not have been set with security in mind. An average 5% of vulnerabilities found over all three years of testing were related to application misconfiguration issues.

6. **Application Privacy Failures.** Related to information leakage/disclosure, application privacy failure occurs when an application has not been properly designed, implemented, or patched, resulting in a potential privacy breach in which an unauthorized user can access data or content. Questions organizations need to ask about privacy in the applications they build include
   - Are our developers trained regarding web application privacy?
   - Are secure coding guidelines being applied?
   - Is our software (including server, database, library code) being kept up-to-date?
   - Are patches being applied regularly?
   - What measures are being taken to ensure that third-party and open source code used in our software is secure and up-to-date?
   - Is personal data being deleted after used for a specified purpose?

   An average 5% of vulnerabilities found over all three years of testing were related to privacy issues.

7. **Insufficient Authentication**. Previously known as Broken Authentication, the OWASP A07:2021—Identification and Authentication Failures category now includes vulnerabilities related to identification failures, which includes both #4 and #7 in our list. An average 5% of vulnerabilities found over all three years of testing were related to insufficient authentication.

   Insufficient authentication and authorization refers to security vulnerabilities in which an application or system does not properly verify the identity of users or fails to enforce proper access controls. Unauthorized users could potentially access sensitive information or execute actions they should not be able to perform, resulting in security issues including data breaches and data loss.

8. **Content Spoofing/Content Injection**. Content spoofing, also known as content injection, is an attack targeting users made possible by a vulnerability in a web application that does not properly handle user-supplied data.

   This can occur when an attacker supplies content to a web application, and that application presents it to unsuspecting users, typically as a modified page under the context of the trusted domain. OWASP notes that this type of attack is widely misunderstood as a common web vulnerability with

Many applications come with developer features that are dangerously unsafe if not deactivated when deployed, such as debug and QA features.

little to no impact. In reality, content spoofing attacks have been used in a variety of scams in which attackers pose as a legitimate entity to trick users into providing login credentials.

Of more concern is that content spoofing has the potential to launch dangerous attacks including code injection and cross-site scripting. An average 4% of vulnerabilities found over all three years of testing were related to content spoofing or content injection.

9. **Susceptibility to Fingerprinting**. Fingerprinting (probing a web application to gain information) can provide attackers with valuable information such as OS type, OS version, SNMP information, domain names, network blocks, VPN points, and more. An average 3% of vulnerabilities found over all three years of testing were related to vulnerabilities associated with fingerprinting.

   Many of the specific fingerprinting security issues the tests uncovered are considered minimal, low, or medium risk. That is, the issues 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, a security issue associated with fingerprinting, verbose server banners (Figure 5), was consistently found in a third of the pen tests and nearly half the DAST scans. Although a medium-risk vulnerability, a verbose server banner could provide attackers with enough information—such as server name, type, and version number—to launch an attack.

	Percentage of Overall High-Risk Vulnerabilities Found	Rank in 2021	Rank in 2020
1. Cross-Site Scripting	19%	2	1
2. Insufficient Authorization	16%	1	2
3. Insufficient Authentication	6%	3	3
4. Application Misconfiguration	4%	6	9
5. Insufficient Transport Layer Protection	4%	4	5
6. Information Disclosure/Information Leakage	3%	7	6
7. Server Misconfiguration	2%	9	8
8. SQL Injection	2%	8	10
9. Unintended Data Leakage	1%	*	*
10. Insufficient Process Validation	1%	*	*

* Not listed in top 10

Figure 3. Top 10 high-risk vulnerabilities found in 2022 (excludes retests)

Of all high-risk vulnerabilities found in the 2022 tests, 19% were found to be associated with to cross-site scripting attacks.

	Percentage of Overall Critical-Risk Vulnerabilities Found	Rank in 2021	Rank in 2020
1. SQL Injection	30%	1	1
2. Insufficient Authorization	9%	2	2
3. Cross-Site Scripting	7%	9	9
4. Insufficient Authentication	7%	3	3
5. Information Disclosure/Information Leakage	6%	4	4
6. Insufficient Process Validation	5%	8	5
7. OS Commanding	3%	7	7
8. Server Misconfiguration	2%	*	*
9. Application Misconfiguration	2%	6	9
10. SQL Injection	1%	*	*

* Not listed in top 10

Figure 4. Top 10 critical-risk vulnerabilities found in 2022 (excludes retests)

SQL command injection was the #1 critical-risk vulnerability found in the tests.

10. **Exposure to Client-Side/Cross-Site Scripting Attacks.** The #1 or #2 high-risk vulnerability found in all three years of Black Duck testing, cross-site scripting (XSS) is a client-side code injection attack. The attacker attempts to execute malicious scripts in a victim's web browser by including malicious code in a legitimate web page or web application. Of all high-risk vulnerabilities found in the 2022 tests, 19% were found to be associated with cross-site scripting attacks (See Figure 3).

    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 as well as data injection attacks (the #1 critical-risk vulnerability found in the tests. See Figure 4). An attacker could use insecure transmissions of user data, such as cookies and forms, to inject a command into the system shell on a web server, and then leverage privileges to compromise the server.

    An insecure or absent CSP might be considered a low-risk concern to many organizations. However, the prevalence of cross-site scripting, clickjacking, and cross-site leak exploits makes a strong argument for having a CSP—and more importantly, a CSP to prevent malicious scripts from executing on the client side—as a second layer of protection against various types of attacks, including cross-site scripting and data injection attacks.

The Danger of Vulnerable Third-Party Libraries

As shown in Figure 6, 25% of the tests conducted by Black Duck in 2022 flagged Vulnerable Third-Party Libraries in Use, correlating with the OWASP Top 10 category A06:2021—Vulnerable and Outdated Components. OWASP notes that your software is likely vulnerable if you do not know the versions of all components you have in use, (both client and server side), including the third-party and open source components your software uses.

	Percentage in Total Test Targets 2022	Rank in 2021	Rank in 2020
1. Weak SSL/TLS Configuration	70%	1	4
2. Missing Content-Security-Policy Header	43%	2	1
3. Verbose Server Banners	37%	3	2
4. Cacheable HTTPS Content	34%	5	5
5. HTTP Strict Transport Security (HSTS) Not Implemented	34%	4	3
6. Insecure Content-Security-Policy Header	31%	6	8
7. Weak Password Policy	28%	7	7
8. Unmasked Nonpublic Information Data	25%	*	*
9. Vulnerable Third-Party Libraries in Use	25%	*	*
10. Excessive Session Timeout Duration	24%	*	*

* Not listed in top 10

Figure 5. Top 10 security issues 2022

Most businesses don't have a good handle on what's in their own code.

Open source code has exploded in popularity and become an essential building block for modern software, as it can dramatically increase the speed and efficiency of software builds. The accessibility and convenience of proven code means that software developers don't have to waste time and limited resources reinventing the wheel.

However, according to the annual Black Duck "Open Source Security and Risk Analysis" report, open source code isn't without risk. In fact, the 2023 report found higher open source security risks than ever before—and that most businesses don't have a good handle on what's in their own code.

The report found that high-risk open source vulnerabilities have increased at a staggering rate over the past five years (557% in the retail and eCommerce space alone). On top of that, there was a disturbing lack of security patching and maintenance of project dependencies (91% included outdated open source components).

In light of high-profile supply chain attacks, software supply chain security has become a major concern for organizations dependent on third-party software, which is literally nearly every modern-day organization. To better manage supply chain risk, more and more organizations are using automated tools to generate a Software Bill of Materials (SBOM) to identify the third-party and open source software they include.

Evidence of the movement toward SBOMs can be seen in the increase of the "create Bills of Materials for software" security activity recorded in the BSIMM report. BSIMM data also indicates significant increases in the "identify open source" and "control open source risk" activities due to the rise of attacks against vulnerable open source projects.

With many companies having hundreds of applications or software systems in use, each themselves likely dependent on hundreds to thousands of different third-party and open source components, an accurate, up-to-date SBOM is urgently needed to effectively track those components.

Insights and Recommendations

- **Implement a multilayered security approach.** Relying on a single security testing solution such as static application security testing (SAST) is no longer sufficient. Organizations should implement a security approach that combines SAST to identify coding flaws, DAST to examine running applications, software composition analysis (SCA) to identify vulnerabilities introduced by third-party components, and penetration testing to identify issues such as misconfigurations as well as vulnerabilities that may be missed by other tests.

  In the Black Duck "Global State of DevSecOps 2023" report, 44% of respondents named external pen testing as a key element of their security testing and a complement to other testing. External pen testing brings benefits such as an unbiased, third-party viewpoint of your security posture, as well as accurate simulation of potential threats and vulnerabilities that other tests might not uncover and that attackers might exploit.

- **Combine automated and manual security testing.** As important as automated testing is for consistency, scalability, and time and cost-savings, the human factor adds a layer of insight and adaptability that is essential for identifying complex and subtle security issues. For example, the very nature of DAST as black box testing (that is, without knowledge of an application's internal structure) requires developer and security experts to verify and triage findings.

- **Prioritize patch management.** Stay vigilant about patching vulnerabilities promptly, especially in third-party and open source components. Establish clear processes to identify, evaluate, and apply patches swiftly to mitigate potential exploits.

The most dangerous vulnerability in your software is the one you're not aware is there.

- **Employ strong access controls.** Enforce policies of stringent access controls across applications and networks. Implement the principle of least privilege, ensuring that users have the minimum level of access necessary to perform their tasks. Regularly review and update access permissions in your applications to prevent unauthorized access.
- **Integrate SBOM generation into your software development life cycle.** The most dangerous vulnerability in your software is the one you're not aware is there. By identifying components with a comprehensive inventory, you can better assess your security status and respond effectively to vulnerability issues when they arise.
- **Determine whether you need to supplement your security testing.** Does your team have sufficient application security skills and resources to test for security defects? Do they have the time to test your software at the level demanded by regulators and your customers?
- **Choose a vendor that can augment your team with on-demand, expert security testing.** Black Duck Security Testing Services can provide continuous access to experts with the skills, tools, and discipline needed to cost-effectively analyze any application, at any depth, at any time.

Black Duck offers a full spectrum of testing services, including penetration testing, dynamic application security testing, static application security testing, mobile application security testing, network penetration testing, red teaming, IoT and embedded software testing, and thick client testing.

Augment your team with our on-demand resources.

Contact Black Duck today to schedule a free consultation.

About Black Duck

Black Duck® offers the most comprehensive, powerful, and trusted portfolio of application security solutions in the industry. We have an unmatched track record of helping organizations around the world secure their software quickly, integrate security efficiently in their development environments, and safely innovate with new technologies. As the recognized leaders, experts, and innovators in software security, Black Duck has everything you need to build trust in your software.

Learn more at www.blackduck.com.