OWASP

An Essential Guide to the OWASP Mobile App Security (MAS) Project

How to build and execute a risk-based security policy using industry standards

NowSecure

Introduction

The world has gone mobile. Adults spend an average of 5 hours and 2 minutes per day in mobile apps, equating to roughly one-third of daily waking hours. Mobile app downloads grew to 255 billion in 2022, up 11% from the previous year.[1]

Together, the Apple App Store and Google Play house some 6 million mobile apps.[2] Given that high volume and release frequency, mobile security and development teams struggle to properly secure their mobile app portfolios.

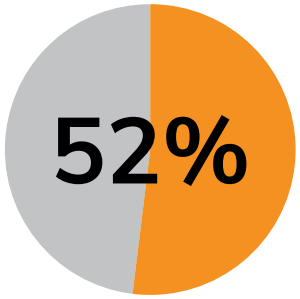
NowSecure has been dedicated to mobile application security for 15 years and our experts assist hundreds of customers in automating mobile app security testing, scaling with continuous testing integrated into the mobile DevSecOps pipeline, verifying security and privacy with mobile Pen Testing as a Service (PTaaS), tapping open-source and commercial pen testing tools and upskilling in secure coding practices. Whether organizations face challenges in developing secure mobile apps, assessing mobile apps or prioritizing remediation, all can benefit from the Open Web Application Security Project (OWASP) global mobile application security standards.

As a committed OWASP God Mode sponsor and the first recognized OWASP MAS Advocate, we wrote this guide to help security and development managers align around standard-based mobile application security testing to reduce risk and speed time to release.

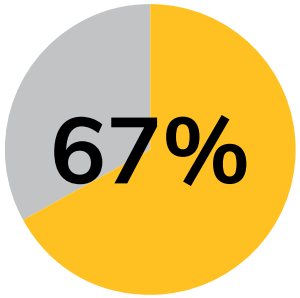
5 hours, 2 minutes

Average daily time spent in mobile apps per user

Source | data.ai



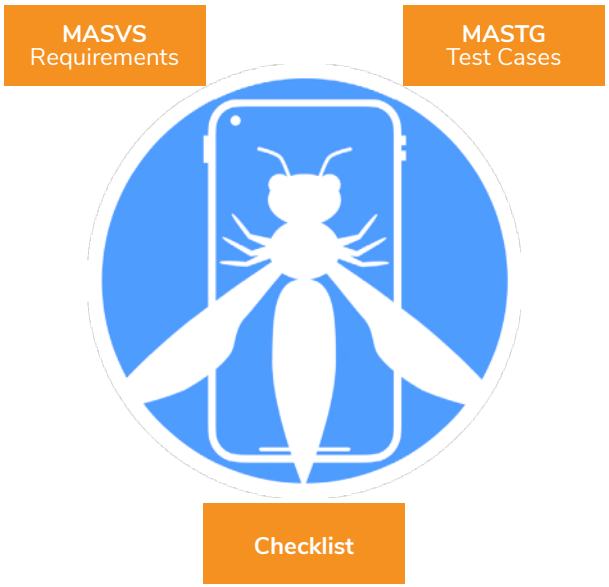
The Verizon Mobile Security Index 2022 found 52% of respondents sacrificed mobile security to get the job done.[3]



Over two-thirds of respondents agreed companies only take cybersecurity seriously enough after they've been compromised.[4]

[1]data.ai, 'Welcome to the Half-Trillion Dollar App Market,' Jan. 18, 2023
[2]Business of Apps, 'App Store Data (2023),' Jan. 9, 2023
[3]Verizon, 'Mobile Security Index 2022,' August 2022
[4]Verizon, 'Mobile Security Index 2022,' August 2022

OWASP Goes Mobile



OWASP has established itself as a highly respected industry standard for web application security. As the popularity of mobile apps grew dramatically, it became apparent that the risks and attack surface for mobile fundamentally differ from web. This mandated a different approach for mobile app security testing.

OWASP launched the Mobile Application Security (MAS) Project for which professionals around the globe contribute to the Mobile Application Security Verification Standard (MASVS), the Mobile Application Security Testing Guide (MASTG), and the Mobile Application Security Testing Checklist.

The MASTG helps security analysts learn how to test for the controls in each MASVS category:

- **V1**: Architecture, Design & Threat Modeling
- **V2**: Data Storage & Privacy
- **V3**: Cryptography
- **V4**: Authentication & Session Management
- **V5**: Network Communication
- **V6**: Platform Interaction
- **V7**: Code Quality & Build Setting
- **V8**: Resilience

Web vs Mobile

Web and mobile applications have vastly different architectures which make them vulnerable to different kinds of risk. Read this white paper to discover why organizations that assess mobile apps with legacy web application security testing tools leave themselves exposed.

OWASP Mobile Application Security Project Resources at a Glance

What is OWASP?

This vendor-neutral open community enables security professionals to contribute unbiased, practical advice about application security. OWASP has become the source that individuals, corporations, universities, government agencies and other organizations look to for worldwide standards in web and mobile app security.

OWASP Mobile Application Security Verification Standard (MASVS)

Description: This resource outlines the different verification requirements for basic mobile app security, defense in-depth app security and reverse engineering resilience.

How to use it: MASVS aids mobile app owners, architects and developers in building security by design, as well as ensuring consistent testing coverage by security professionals.

See the OWASP MASVS

OWASP Mobile Application Security Testing Guide (MASTG)

Description: This is an ever-growing manual for security analysts to test MASVS.

How to use it: MASTG is a reference guide for security analysts of all levels of experience to promote well-rounded mobile app security testing.

See the OWASP MASTG

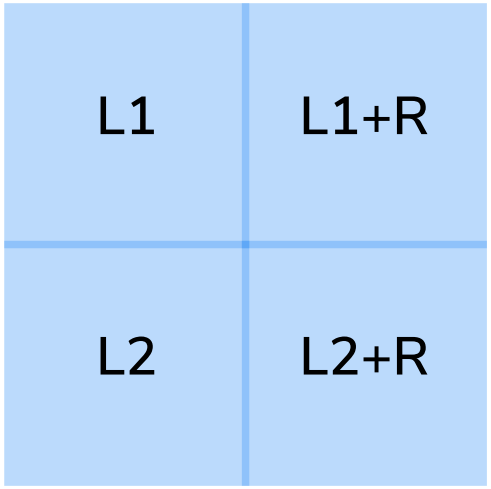
OWASP Mobile Application Security Testing Checklist

Description: This spreadsheet outlines items to pass or fail for MASVS requirements and references sections of the MASTG to assist with testing.

How to use it: Security analysts can use the checklist as a template for management reports when performing mobile app assessments.

See the OWASP Mobile Application Security Testing Checklist

Mobile AppSec Threat Modeling Using OWASP MASVS



Above: The four different levels of security verification using OWASP MASVS

Ensuring all mobile app stakeholders understand the app risks can help everyone make decisions about vulnerabilities. Organizations should apply one of four different levels of security verification per mobile app depending on the characteristics of the app as shown in the diagram to the left.

L1 – Standard Security Verification

L1 describes the most basic level of security verification any mobile app should undergo. This might be a basic calendar application, a voice memo app or an event app with high-level event information. These apps do not handle intellectual property, critical transactions or user information that is considered high risk.

L2 – Defense-in-Depth Verification

L2 requires L1 tests and additional defense-in-depth controls, meaning the mobile app must be resilient to more sophisticated attacks. Mobile apps that reside in this quadrant tend to interact with highly sensitive data such as credit card or healthcare information that can be used for fraud.

L1 + R – Standard Security + Reverse Engineering Resiliency

R stands for the reverse engineering resiliency requirements. These are critical to preserving the integrity of mobile app functionality and protecting confidential information such as intellectual property. A gaming app, for example, requires L1 testing with reverse engineering resiliency. Mobile app game developers cannot afford to allow users to tamper with the app and cheat the system. Anti-tampering requirements protect the integrity of game play.

L2 + R – Defense-in-Depth + Reverse Engineering Resiliency

Online banking apps provide a prime example of apps needing L2 verification with reverse engineering resiliency. Essentially, if the app handles sensitive data, performs high value transactions, needs to comply with regulatory regimes and/or must prevent leakage or tampering of personally identifiable information (PII), the app likely needs L2 + R level of verification.

Mobile AppSec Threat Modeling Using OWASP MASVS

MASVS L1
Standard Security

- This is the baseline
- No compliance or regulatory needs
- No intellectual property (IP) or highly sensitive data handled
- No high value transactions performed
- Simple apps

MASVS L1+R
Standard Security
+ High RE Resilience

- Prioritize IP protection
- Prevent malicious modification or tampering

MASVS L2
Defense-in-Depth

- Regulated industry data
- Compliance consideration
- Apps that perform simple tasks, but handle highly sensitive data

MASVS L2+R
Defense-in-Depth
+ High RE Resilience

- Apps that perform complex activities between users and handle highly sensitive data
- Compliance and IP protection are key
- Prevent malware-based attacks

In order to understand which quadrant of verification a mobile app requires, consider the purpose of the app and how it will be used. Organizations need to consider three key areas:

- **Purpose:** What does the app do?
- **Data:** What data does this app handle?
- **Perspective:** Protecting the organization is important, as is complying with regulatory standards, but let's not forget that user demand and experience drives mobile app success.

To help make these different levels of verification concrete, let's step through the domains V1-V8 of MASVS and tie security vulnerabilities and business impact stories to each domain. While domain V1 focuses more on the design and architecture of the app and not specific test cases, domains V2-V8 require static analysis (SAST) and dynamic analysis (DAST) to thoroughly exercise the app and confirm it meets the security requirements listed in MASVS.

V1: Architecture, Design and Threat Modeling

This domain focuses on the architecture and design of the mobile app. Architects, development leaders and mobile app owners should take these requirements into consideration when designing and building mobile apps.

MASVS requirement below and consider the ramifications in the real-world example that follows.

MASVS Requirement in Question

Security Requirement 1.5

All app components are defined in terms of the business functions and/or security functions they provide.

Sample Context

A digital wallet that lets friends share payments skyrockets to popularity for its ease of use. This app handles user personally identifiable information (PII), banking information, has high visibility and high impact to customers, and the code functionality of the app should be tamperproof. Therefore, it requires L2+R type of security verification.

The Attack Scenario

A young man walks into a bar and strikes up a conversation with a young lady. Things are going well so he asks for her number and hands his phone over for her to enter it. But instead of entering her number, she transfers $2,500 to herself using his digital wallet. Because the app doesn't enable session management by default, he was unable to proactively prevent the attack. She hands the phone back to him, says goodbye, and walks away $2,500 richer.

The Business Impact

While con artists and swindlers have been tricking lovers for centuries into giving away sums of money, this digital wallet is just another play in their bag of tricks. The digital wallet company loses millions of dollars from scams, fraudulent payments, and ultimately customers who will never use its mobile app again.

OWASP MASVS Reference

V1: Architecture, Design and Threat Modeling Requirements

OWASP MASTG Reference

N/A - Because this domain does not have specific test cases, there is no reference to the testing guide.

V2: Data Storage and Privacy

As privacy concerns grow, organizations must protect user data more proactively in even the most basic mobile apps. Data classified as “sensitive” may vary across organizations. Sensitive data may include location data, username and passwords, personally identifiable information (PII) and documents with confidential intellectual property (IP).

The example below maps a finding to the V2 domain of MASVS. Links to MSTG are referenced in the sidebar.

Sample Finding
Sensitive Data (PII) Found in Public Device Storage

Brief Description of Finding
The application saves sensitive data to a public app folder on the device that’s accessible by other apps.

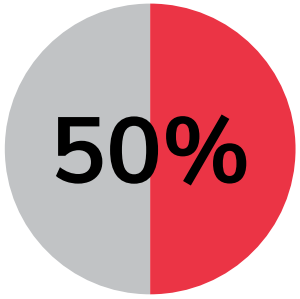
MASVS Requirement in Question
Security Verification Requirement 2.2

No sensitive data should be stored outside of the app container or system credential storage facilities.

Sample Context
A healthcare provider invests $800,000 to build a mobile app for users to track vital protected health information (PHI) to inform them when to take their medicine and control updates to their Internet of Things (IoT) medical devices. It believes this initiative is mandatory for the best user experience. This app handles user personally identifiable information (PII), has high visibility and high impact to customers, and the code functionality of the app should be tamperproof. Therefore, it requires L2+R type of security verification.

The Attack Scenario
An attacker discovers that this app stores data and firmware updates in a public app folder on the device. This is considered unprotected storage and can be accessed by other applications. Firmware updates can be modified and wreck the user experience, or worse, seriously compromise the user health.

The Business Impact
Because this IoT device has become woven into the lifestyle of its users, disruption in usage can range from mildly annoying to life threatening. The risks include litigation and tarnished brand reputation. User adoption is critical to revenue stream. Privacy gaps and firmware malfunctions would significantly impede IoT adoption and company growth.



of mobile apps tested by NowSecure software **store data insecurely**. Frequently discovered unencrypted PII include username, account number, phone number, IMEI, GPS location, Wi-Fi/ MAC address and more in system files and local files on-device.

OWASP MASVS Reference
V2: Data Storage and Privacy Requirements

OWASP MASTG Reference
- Android Data Storage
- iOS Data Storage

V3: Cryptography

Cryptography is essential to protecting sensitive data. The requirements in this domain relate to correct implementation of cryptography.

Sample Finding
Hardcoded values used for crypto, weak algorithm for crypto.

Brief Description of Finding
The application was found to be using hardcoded values to generate cryptographic data. It was also discovered that the app relies on an outdated cryptography method, AES-ECB, to generate cryptographic data.

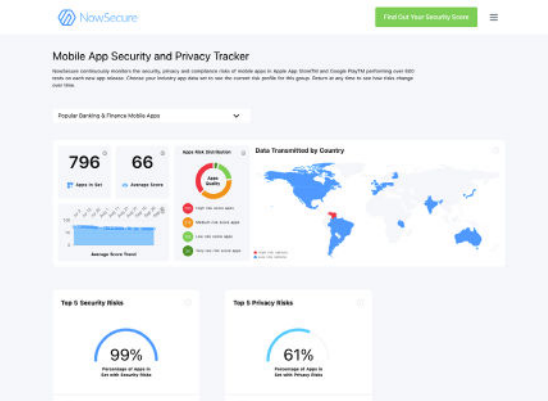
MASVS Requirement in Question
Security Verification Requirement 3.2

The app uses proven implementations of cryptographic primitives.

Sample Context
A secure messaging app handles highly confidential intellectual property (IP), has high visibility and high impact to users, and the code functionality of the app should be tamper-proof. It requires L2+R type of security verification from both the owner and user perspective.

The Attack Scenario
The app attempts to prevent attackers from obtaining their customers' messages in every attack scenario, so it implements client-side cryptography to protect said data. Unfortunately, a weak algorithm was employed and the key used to generate encrypted data was exposed within the app.

The Business Impact
A secure messaging app that turns out to be leaky would seriously harm the brand reputation and revenue stream of the app producer. Additionally, the company would probably need to cover the damages incurred to its customer that lost IP. The customer would also suffer a loss of brand reputation and first-to-market advantage, among other negative consequences.



Check real-time security and privacy issue benchmarking for popular mobile app categories with the NowSecure MobileRiskTracker.

OWASP MASVS Reference
V3: Cryptography Requirements

OWASP MASTG Reference
- Mobile App Cryptography
- Android Crypto
- iOS Crypto

V4: Authentication and Session Management

When users are granted access to a remote service, the endpoint must confirm the users' identity and privileges for the remote service. Additionally, the remote endpoint should employ rate limiting to protect against brute-force attacks.

Sample Finding
Denial of services due to lack of rate limiting

Brief Description of Finding
The application does not implement rate limiting during registration of physical hardware, leading to a denial-of-service attack.

MASVS Requirement in Question
Security Verification Requirement 4.6

The remote endpoint implements a mechanism to protect against the submission of credentials an excessive number of times.

Sample Context
A children's toy manufacturer invests heavily to revamp a retro favorite for the modern age. The popularity of a new TV show, comic and movie have driven sales of the new IoT toy through the roof — it is the hottest toy of the holiday season. This app requires L2 security verification because it requires user information for authentication and must comply with the Children's Online Privacy Protection Act (COPPA).

The Attack Scenario
Toy owners use an app to interact with it, but they first must create an account and register the serial number. This registration process connects the toy to the owner's account. An attacker realizes the serial numbers are not only short and sequential, but also discovers that the endpoint lacks rate limiting. He attempts a brute-force attack against the endpoint and registers more than a million serial numbers yet to be claimed.

The Business Impact
Thousands of eager new toy owners attempt to create their accounts on the mobile app only to discover their toy has already been registered to another account. Consequently, these children cannot interact with their toy as was marketed. Parents return the defunct toy and take to social media to air complaints about poor customer support and functionality. The toy manufacturer's stock price plummets and its brand reputation suffers from such a high-profile failure.

OWASP MASVS Reference
V4: Authentication and Session Management Requirements

OWASP MASTG Reference
- Mobile App Authentication Architectures
- Android Local Authentication
- iOS Local Authentication

V5: Network Communication

This domain focuses on ensuring the confidentiality and integrity of mobile app data transmitted over the network. Developers should use encrypted channels for communication using the TLS protocol and in some cases, certificate pinning, to safeguard data in transit.

Sample Finding
Hostname verification improperly implemented

Brief Description of Finding
The application validates the certificate authority (CA), but not the hostname.

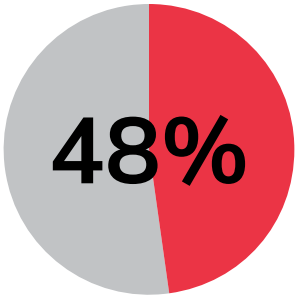
MASVS Requirement in Question
Security Verification Requirement 5.3

The app verifies the X.509 certificate of the remote endpoint when the secure channel is established. Only certificates signed by a trusted CA are accepted.

Sample Context
A large retail company invests over $1 million to build a mobile app. Based on an analysis of its customer base, the company projects the initiative will lead to a 10% increase in annual sales by driving more customer engagement and transactions. Because this app handles userpayment information, the network functionalities in the app should be protected by certificate pinning. This app requires L2 type of security verification from the user perspective and L2 + R from the owner perspective.

The Attack Scenario
In a rush to meet deadlines and jumpstart customer adoption, the mobile app gets pushed to the app store with no security testing. An attacker sees that the app not only failed to implement certificate pinning, but the hostname verification was not properly implemented, and takes advantage of it to launch a phishing attack. Not even a Mobile Device Management system or Virtual Private Network can prevent this attack.

The Business Impact
This phishing attack tricks customers into sending money abroad and creates a social media storm that damages brand reputation and customer loyalty. Adoption of the mobile app stalls, customer transactions dip, stock prices drop, and by the end of the quarter, overall company earnings dip 3%. This doesn't even take into account the ensuing lawsuit and fines.



of mobile apps tested by NowSecure software have **insecure network communications** frequently caused by improper hostname verification and improper certificate validation, which expose users to malicious endpoints, data theft and phishing attacks.

OWASP MASVS Reference
V5: Network Communication Requirements

OWASP MASTG Reference
- Mobile App Network Communication
- Android Network Communication
- iOS Network Communication

V6: Platform Interaction

These controls ensure that the app uses platform APIs and standard components in a secure manner. They also address communication between apps.

Sample Finding
Native objects exposed through webviews

Brief Description of Finding
The application exposes Java objects through the use of a webview leading to remote code execution through a Javascript interface.

MASVS Requirement in Question
Security Verification Requirement 6.7

If native methods of the app are exposed to a WebView, verify that the WebView only renders JavaScript contained within the app package.

Sample Context
A weather app uses Webviews to generate ads within the user interface (UI) and drive revenue for the app developer. This app doesn't handle any sensitive data other than location data and it doesn't need to be tamper proof, so the app requires L1 type of security verification.

The Attack Scenario
This app not only uses addJavascriptInterface, it also uses HTTP requests to render UI content within the app from javascript and HTML. An attacker executes a man-in-the-middle (MiTM) attack to inject malicious Javascript into the app, phishing users for personal information and trolling users with unwanted images. The native bridge created by the addJavascriptInterface allows the attacker to generate Toast notifications in the users device, and the use of HTTP loaded by a webview allows him to render any content within the app he desires.

The Business Impact
Jarring inappropriate images would abruptly interrupt usage of the app and drive users to uninstall the mobile app. In addition, the Javascript interfaces would allow an external entity to interact with the native code within the app. In some cases this could be as simple as showing toast notification in the UI, and at worst, it could be used to send requests to other apps on the device. This would impede the app developer's revenue stream because there are plenty of other weather apps on the market.

OWASP MASVS Reference
V6: Platform Interaction Requirements

OWASP MASTG Reference
- Android Platform APIs
- iOS Platform APIs

V7: Code Quality and Build Setting

The purpose of this control is to ensure developers follow basic security coding practices in developing the app and take advantage of security features built into the compiler.

Sample Finding
AFNetworking vulnerability detected

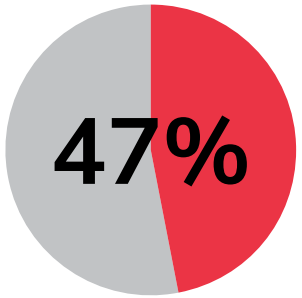
Brief Description of Finding
The application was found to be using an outdated version of the AFNetworking library. This vulnerability was patched as of version 2.5.2. However, if an older version is used, it allows all the SSL traffic to be intercepted and decrypted in a standard MiTM environment.

MASVS Requirement in Question
Security Verification Requirement 7.5
All third-party components used by the mobile app such as libraries and frameworks are identified and checked for known vulnerabilities.

Sample Context
A developer of a dating app uses code from an old code base containing an outdated version of AFNetworking framework. The code is included in a new feature update that allows the user to display his full profile info, including address and phone number from the app UI. Because of a short deadline, the code is pushed to production without security testing. This app does not have regulatory requirements, so it requires a basic level of security verification (L1).

The Attack Scenario
Similar to network communication issues in domain V5, an attacker can exploit this issue via a MiTM attack. The core difference from the perspective of the developer is he has done everything correctly, but used a vulnerable version of the AFNetwork framework.

The Business Impact
Without any security testing, the app was deployed and the vulnerability was launched into the wild. A security researcher posts a Twitter tirade about how the app should never be used for privacy concerns. The uninstall rate soars and the company's stock plummets.



of mobile apps tested by NowSecure have **insecure exploitable extraneous functionality**. In many instances, these issues lie in thirdparty libraries. Proper implementation of the latest versions of code libraries is critical to code quality. Common web issues like SQLInjection and XSS occur in mobile apps as well.

OWASP MASVS Reference
V7: Code Quality and Build Setting Requirements

OWASP MASTG Reference
- Mobile App Code Quality
- Android Code Quality and Build Settings
- iOS Code Quality and Build Settings

V8: Resilience

Defense-in-depth tactics seek to increase resilience against unauthorized tampering, reverse engineering and specific client-side attacks. Resiliency is required for apps that process or grant access to sensitive data and functionality.

Sample Finding
Lack of anti-tamper techniques

Brief Description of Finding
The app does not implement anti-tamper techniques to prevent malicious modification of the code.

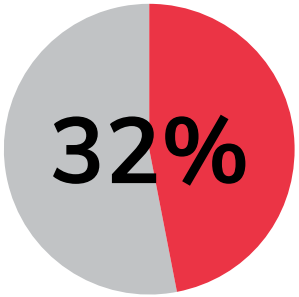
MASVS Requirement in Question
Security Verification Requirement 8.6
The app detects and responds to tampered code and data in its own memory space.

Sample Context
A retail app implements a new coupon process to increase holiday sales, allowing customers to show their "single use" coupons in-app while shopping at the store or online. The convenience is a huge success with an initial spike in app downloads and increased sales.

The Attack Scenario
The app implements good cryptography practices, certificate pinning, and other security measures, including expensive industry-standard app protection technology. While these measures help protect the user's security and privacy, there is still a major issue within the app's logic.

Unfortunately, the coupons are generated on the client side without tamper protection, allowing an attacker with a penchant for new electronics to reverse engineer the app and generate coupons of their own decided criteria on the fly. Not only are these coupons absurdly good deals, but also are valid in the eyes of the payment system.

The Business Impact
The attacker shares how to use the exploit, leading to massive deals on the most expensive items in the store. Unfortunately, the retail company's slow remediation process doesn't close this gap quickly enough, infringing on holiday sales.



of mobile apps tested by NowSecure have **exposure to reverse engineering**. iOS has strong DRM features built in to improve resilience, but Android app developers need to build or license more resilience features due to the Java foundation.

OWASP MASVS Reference
V8: Resilience Requirements

OWASP MASTG Reference
- Android Anti-Reversing Defenses
- iOS Anti-Reversing Defenses

Standardize. Scale. Share.

Mobile app risk continually evolves as mobile platforms, operating systems and development environments advance.

When you scale mobile app security testing, use this as a reference guide to frame the challenge of securing an ever-growing mobile app portfolio with finite resources. Start standardizing mobile app security testing by triaging your current mobile app portfolio. Then build a risk-based security policy for mobile apps to scale as needed.

No single analyst, security expert, developer or architect can solve mobile app security problems alone. It takes a village to deliver secure mobile apps. And it takes great communication and collaboration to speed the production of secure mobile apps.

Share this guide with your colleagues. Establish a common language and framework with them. Create processes that enable mobile app stakeholders to make swift decisions without compromising the security posture of your organization. And most importantly, give back to the community. Global security professionals contribute to OWASP projects and continue to improve the library of resources for mobile app security. We're all in this together to save the world from unsafe mobile apps.

Free OWASP Education

Learn more about OWASP Mobile App Security (MAS) and explore mobile security by enrolling in free courses from NowSecure Academy. The online training portal includes modules on OWASP mobile vs web standards, the OWASP MAS and forthcoming updates to the MASVS and MASTG.

Enroll in NowSecure Academy today

NowSecure mobile app security solutions feature OWASP at the very foundation:

Standardize, scale and demonstrate MASVS compliance with NowSecure Platform automated mobile app security testing.

Conduct pen testing with the NowSecure Workstation toolkit.

Verify the security of high-risk mobile apps and demonstrate MASVS compliance with NowSecure Mobile Pen Testing as a Service (PTaaS).

Upskill on secure coding practices and mobile security via NowSecure Academy.

Contact NowSecure for a private briefing or have one of our mobile experts attend your local OWASP meetup event.

NowSecure offers a comprehensive suite of automated mobile app security and privacy testing solutions, penetration testing and training services to reduce risk. Trusted by many of the world's most demanding organizations, NowSecure protects millions of app users across banking, insurance, high tech, retail, healthcare and government. The company is SOC2 certified and was named a mobile security testing leader by IDC and a DevSecOps transformational leader by Gartner. Visit www.nowsecure.com to discover strategies for strengthening security and speeding the development of high-quality mobile apps.