

FORTRΔ

Understanding Mobile Application Penetration Testing



Table of Contents

<u>BENEFITS OF MOBILE APP PEN TESTING</u>	5
<u>WHO NEEDS MOBILE APP PEN TESTING</u>	7
<u>COMMON MOBILE APP VULNERABILITIES</u>	9
<u>METHODOLOGY AND STANDARDS</u>	11
<u>MOBILE APP PEN TEST PARAMETERS</u>	13
<u>MOBILE APP PEN TESTING IN 3 EASY STEPS</u>	15
<u>MAKING MOBILE APP PEN TESTING PAR FOR THE COURSE</u>	18



[Mobile apps](#) are nothing if not convenient. Roughly [62%](#) of businesses either have an app or are on their way to developing one, and last year, over [60% of](#) internet traffic came from mobile sites (up from 29% ten years ago). Mobile apps get information to customers faster and facilitate the online buying process, likely resulting in [more sales](#). And on the operations side, enterprises themselves use [over 1000 mobile applications](#) in their course of daily business.

The Cost of Convenience

Unfortunately, the security cost of mobile app convenience can be exceptionally high. Let's look at the [stats](#).

- Business apps are three times more likely to give up credentials than the average application.
- Over 75% of all applications have at least one vulnerability.
- One in every four mobile apps is burdened with a security flaw.

Mobile devices (and the applications they host) can be primary endpoints, which attackers use to infiltrate a network. Mobile app source code, as part of an often-murky supply chain, can come from anywhere and carry virtually anything. Even ones created in-house can be subject to oversights and flaws.

Protecting Mobile Endpoints

The above figures don't bode well for mobile application security optimists, but it does paint a picture for the realists. The numbers simply reflect a starting point. Using mobile application penetration testing, organizations can detect application security flaws before the threat actors do, giving teams a chance to remediate them and stay safe. A vulnerability might be inevitable at these rates, but a successful attack at the endpoint doesn't have to be.

Here's how mobile app pen testing can help teams vet their mobile applications for security flaws and shut down a major point of entry into their network.

**Mobile app source code
can come from anywhere
and carry virtually
anything. Even apps
created in-house can be
subject to oversights and
flaws.**



Benefits of Mobile Application Penetration Testing

Mobile application penetration testing is the offensive security practice of identifying vulnerabilities within your iOS or Android application by simulating the same attack procedures as a real-world threat actor. As mobile apps became more ubiquitous in the early 2000s, mobile app pen testing gained traction. The benefits of Mobile Application Penetration Testing include:

- 1. Security Feature Validation** – Ensuring the security features provided by the app, such as biometric authentication, are being used.
- 2. Securing API Integrations and backend components:** With [71% of all web traffic](#) being attributed to API calls now, it is clear that web security (and app security by extension) means a focus on securing API and backend components.
- 3. Maintaining Industry Compliance Standards:** Pen testing is a required or implied requirement of several, including PCI DSS.
- 4. Preserving Uptime:** When Amazon experienced [59 minutes of downtime](#) in 2021, they missed out on \$34 million in sales. Large organizations can lose up to [\\$9,000 per minute](#) in outages, and for context, the top-earning apps can pull in about [\\$35,000 per day](#). That's a hefty sum, but it can be cancelled out quickly when services get knocked offline due to malicious attacks.
- 5. Maintaining a Competitive Edge:** There are nearly 3 million apps available for download on the Google Play store and nearly [2 million](#) on the Apple App Store. The message is clear: If a customer doesn't like yours, for whatever reason, there are plenty of other fish in the mobile application sea. Overwhelmingly, customers are valuing security in their

mobile apps; a recent survey by AppDome revealed that "95.5% – the highest level ever" demanded comprehensive mobile protection, including "[mobile app data](#) [protection], account integrity, login, data storage, data in transit, and protection from malware and fraud."

Ultimately, pen testing your mobile app gives you a higher chance of staying in the game; with customers, with uptime, with compliance, and in a highly competitive – and highly targeted – arena.

TLDR: Mobile Application Penetration Testing evaluates the security of mobile app code and backend components, identifying potential attack paths so they can be remediated before they are exploited.

Who Needs Mobile App Pen Testing and Why

Based on the ubiquity of mobile applications in today's workforce, it would seem nearly everyone needs mobile application penetration testing. Considering the criticality of apps and the current state of mobile app security, it is easy to see why.

Endpoint security boasts a nearly 13% CAGR and is projected to grow from \$14 billion to [\\$26 billion](#) in the next five years. Mobile devices are [one of only two segments](#) in the unified endpoint management market (outnumbered only by desktops), and with an ever-growing BYOD culture, they may even become the primary one in years to come. The making of a mobile app is often a multi-pronged process, and the pieces can be difficult to track, much less subject to uniform security standards during the build.

[Did you know?](#)

- 79% of SMBs outsource their mobile app creation to an app development team or a freelancer.
- 43% of small businesses build their mobile apps using DIY software.
- As of two years ago, a full 97% of applications used [open source software](#).

Because there so many variables and so varied parties involved, Mobile apps should never be taken at face value. Organizations should make it a practice to vet them out-of-the-box (even if it's your box) before putting them into circulation (and at regular intervals thereafter).

TLDR: Any organization will benefit from mobile application penetration testing, as mobile applications are ubiquitous. Additionally, the source of the app is irrelevant. Whether it is homegrown or from a third party, security weaknesses could be present and need to be proactively identified.

The background of the slide features a dark, semi-transparent image of two men in a professional setting. One man, wearing a plaid shirt, is looking at a laptop while the other, in a hoodie, looks on. The image is overlaid with a pattern of overlapping triangles in various shades of green and blue. The title text is overlaid on the bottom left of this image.

Common Mobile App Vulnerabilities

[The Open Worldwide Application Security Project \(OWASP\)](#) is a non-profit whose sole focus is to improve the security of software for businesses, developers, and customers. Aligned with that goal, the [OWASP Mobile Top 10](#) is a list of the most critical vulnerabilities and security risks developers face when spinning up mobile applications. Released every year, the most recent line-up included:

- 1. Improper Credential Usage** | Attackers exploit hardcoded credentials with publicly available tools.
- 2. Inadequate Supply Chain Security** | An attacker can undermine the mobile app by introducing spyware, backdoors, or other malicious code via modifications in the build process, the mobile app codebase, or in third-party libraries.
- 3. Insecure Authentication/Authorization** | Threat actors typically leverage automated attacks to exploit authentication and authorization vulnerabilities.
- 4. Insufficient Input/Output Validation** | By not validating and sanitizing external data, mobile apps are susceptible to attacks such as SQL injection, command injection, and cross-site scripting.
- 5. Insecure Communication** | When mobile devices exchange data with remote servers, threat actors can listen in on the wire and intercept messages. They might get their access via monitored or compromised Wi-Fi, rogue network devices, or malware.
- 6. Inadequate Privacy Controls** | Because PII is typically well protected, attackers often resort to non-traditional methods of access such as leveraging a trojan to access logs, eavesdropping on network communications, or physically steal the mobile device and create a backup for analysis.
- 7. Insufficient Binary Protections** | App binaries are targeted for their valuable secrets, critical business logic, or position in larger supply chains using reverse engineering or code tampering.
- 8. Security Misconfigurations** | Mobile app security misconfigurations can be exploited via improper access controls, internal default settings, weak hashing or encryption, and more.
- 9. Weak APIs** | APIs act as the hubs that let apps integrate with other apps, services, and widgets, but strong authentication, encryption, and access controls are needed to secure them against outside attackers.
- 10. Insecure Data Storage** | Attackers compromise weak mobile data storage to pilfer sensitive financial information, personal data, IP, and information used for cyber espionage, among other things. Common vectors include illicit physical or remote access, malware, rooted or jailbroken devices, and social engineering techniques.
- 11. Insufficient Cryptography** | Weak key length encryption, vulnerabilities in cryptographic frameworks, and insecure hash functions can lead to compromises like the reverse-engineering of hashed data, data manipulation, and unauthorized access.

TLDR: From improper credentials use and weak APIs to anemic privacy controls, there are several vulnerabilities commonly seen in mobile apps that can be quickly discovered and eliminated through mobile application penetration testing.



Methodology and Standards

While there are some mandatory security standards that mobile apps must clear before hitting the app store, there are more thorough regulations that better prepare you to maintain a competitive level of security for your enterprise-level environment. By adhering to these stronger external security standards, your mobile applications are more likely to earn the confidence of customers, partners, and prospects.

Here are three significant ones.

NIAP

The National Information Assurance Partnership's (NIAP's) standards for mobile application security testing are based largely on its [Protection Profile for Application Software](#) (All PP), which "defines the security requirements that need to be met by application software that runs on mobile devices, desktops, and servers." In 2020, the NIAP came out with [automated testing guidelines for vetting mobile](#) applications which allows these complex processes to be done at scale.

ioXt MAP

Touted as "[The Global Standard for Mobile App Security](#)," the ioXt (Internet of Secure Things) Mobile Application profile (MAP) is a security framework that "applies to any cloud connected mobile app and provides the much needed market transparency for consumer and commercial mobile app security."

OWASP MAS (MASVS & MASTG)

The mission of OWASP [Mobile Application Security](#) is nothing less than to "define the industry standard for mobile application security." To do so, this 'flagship' project lays out both a security standard for mobile apps (MASVS) and a comprehensive guide outlining how to properly test mobile applications against that standard (OWASP MASTG).

The [Mobile Application Security Assessment](#) (MASVS) "provides a set of baseline security criteria for developers" and the [Mobile Application Security Testing Guide](#) (MASTG) acts as a published set of testing criteria. Google uses MASA to recognize developers who have had their applications pass MASVS Level 1 requirements.

95% Fail to Pass Security Standards

By adhering to external standards for the safe production of mobile apps, organizations can assure their customers, clients, and partners that the applications coming from them will be a security asset, not a liability. However, these frameworks are optional, and unfortunately, many opt out. Mobile security firm NowSecure revealed that a shocking 95% of popular mobile apps out there fail to pass the OWASP MASVS standards, for example. The standards help ensure security – but only insofar as they are applied.

For this reason, organizations would be well served to implement alternative security routes such as penetration testing. By recognizing where their app security falls short, they can bolster their defensive posture and meet the external frameworks (like MASVS, MAP, and NIAP) that will give them a competitive edge in an increasingly security-conscious world.

TLDR: A mobile app is only as good as its security, so it's imperative organizations work towards the highest security standards, vs. just checking the boxes on an easy path to market. Those who skimp on security will pay the price through downtime and/or breaches and quickly lose ground to their competitors.



Pen Testing Parameters

When putting your mobile application through its paces, you want to test what matters. Here are the essential elements that underpin a mobile app's functionality and which, if compromised, could lead to its downfall.

- **Code:** You want to vet the app's underlying code and architecture, both before and after it rolls off the line (SAST and DAST tools are especially good for this).
- **Data storage:** Next, look at how the app stores data. Understanding the data storage mechanism – and the security policies that may or may not protect it.
- **Network connectivity:** Assess how the mobile app communicates over the network and identify potential security risks in those communications. This includes TLS configuration, certificates, API endpoint security controls, etc.
- **Authentication methods:** Weak, stolen, or reused passwords account for [81% of breaches](#), and apps are no exception to poor authentication. Mobile pen testing can test authentication using SQL injection, brute-force attempts, session hijacking, credential stuffing, and even social engineering.
- **APIs** | Vet for API openings where attackers could find their way through. Shadow APIs account for [30%](#) of all malicious API attacks, and finding one unattended (and unprotected) could act as a gold mine into the app's most sensitive function. It could also serve as a pivot point into other connected apps, making your application the compromised link in a supply chain attack. Even a known API could facilitate this kind of trouble if not secured correctly.

- **User interface (UI):** Vulnerabilities in the app's surface – its UI – can lead to breaches via deleted information, injection attacks, broken access controls, URL tampering, form altering, and manipulated design flaws that get users to engage with fraudulent elements of the UI and give away sensitive data.

TLDR: A mobile application pen test must be comprehensive in its scope; leaving an element out could mean leaving a blind spot in your security strategy – and an open door to attackers.



Mobile App Pen Testing in 3 Steps

Testing your mobile applications before they hit public consumption is certainly best practice. However, customers have been trained to expect updates, so there is no shame in getting to it now if your application is already on the market. Here are the three essential steps to successfully penetration testing your mobile application – no matter what phase it is in. Attackers don't give up once your product rolls off the line, and neither should you.

Step 1: Analysis

This essential preliminary step of analysis prepares you to strategize your best offensive attack once you launch into the exploitation phase of your mobile application pen test. It enables you to [assess all the latent security risks](#) that might be hiding within your application's codebase, your network traffic, your APIs, and your digital footprint at large.

IAC Analysis: Software-defined infrastructure, or Infrastructure as Code (IaC), "allows the configuration and deployment of infrastructure components faster with consistency by allowing them to be defined as a code and also enables repeatable deployments across environments," according to [OWASP](#). IaC scanning checks for vulnerabilities in this "code" and runs in your CI/CD pipeline. If you can single out vulnerabilities before they become solidified as a default, you can prevent their spread within your network.

OSINT: Next, leverage Open Source Intelligence (OSINT) to perform the kind of basic reconnaissance an attacker would when scoping out your network. OSINT includes gathering data from across publicly available sources like social media, online databases, websites, and professional platforms in order to get the most (and best) information with which to launch your attack. This can include IP addresses, contact information, employee details, technology stacks, and more – anything that will give you leverage into an area of potential weakness.

File Systems: Mobile App Pen Testing also requires you to test the application's file for vulnerabilities. These can vet for insecure data handling practices, potential exploits, and source code flaws relating to data encryption, invalidated file uploads, file handling, and improper permissions leading to unauthorized file access. Both static and dynamic application testing can be applied here.

Reverse Engineering: Reverse engineering is a critical task in the context of mobile application penetration testing, as it involves analyzing the underlying code and structure of the application to identify potential vulnerabilities or security weaknesses. This process typically starts with extracting the application package and decompiling it into readable code. The primary goals include understanding the app's logic, uncovering sensitive hardcoded information such as API keys or credentials, and identifying potential flaws in obfuscation or encryption mechanisms.

Network Traffic: Much can be gleaned from examining the network traffic generated by the app. This can help identify data transfer protocols (HTTPS vs. HTTP), the transmission of sensitive data, and any endpoints with which the app is communicating. Data like this can provide invaluable clues for further pen testing strategies down the road.

The more thorough your analysis and reconnaissance phase, the more fun and effective your exploitation stage will be.

Step 2: Exploitation

Now, for the fun part. With the end goal in mind – to discover just how far a threat actor could run with these discovered vulnerabilities and weaknesses – it is time to simulate a real-life attack against your mobile application.

How you do the actual work of exploitation is up to your team. You can create your own custom exploits based on specific

vulnerabilities inherent to your app. Or you could use pre-built exploits designed to target vulnerabilities common within most mobile applications.

You could do this portion of the testing in-house if you have a standing penetration testing team. [Automated penetration testing software](#) can help your standing staff use the same exploits as today's attackers, complete with guided walk-throughs suited to any level of analyst. These tools can walk your team through an entire engagement or just automate the more routine aspects of pen testing so your analysts can focus on diving even deeper.

Or, if you are more inclined to budget resources, you could opt for a third-party service provider to run the [pen testing services](#) for you, from reconnaissance to report. This enables you to keep your staff handy for any immediate security needs and lessens the burden on your workflow, schedule, and expertise.

Whichever way you choose, the exploitation phase is where the rubber hits the road, and all your theories are proven. Although certain weak points and vulnerabilities may have been discovered during the analysis phase, it is not until they have been attacked and exploited that your team will understand which are fairly harmless and which should be immediately prioritized.

Step 3: Reporting

Lastly, a pen test isn't official without a pen test report. The report is, arguably, the purpose behind the whole penetration testing exercise in the first place, as it proves the findings and validates the results. Effective reports not only prove that vulnerabilities were discovered but demonstrate the ultimate impact of their possible exploitation on the organization.

It builds the business case for leaders to invest in more security spend in order to allay these consequences, and lets the facts "speak for themselves," rather than giving security practitioners an uphill battle. Now, requests for greater funding and initiatives will be backed by hard data, and SOCs will have their work clearly defined until the next pen test. Mobile App Pen Testing prioritizes the discovered vulnerabilities based on severity, so strapped teams know where to rally and remediate first.

Effective Mobile App Pen Testing reports provides detailed assessment of the security posture of the mobile application. It gives the security product team feedback that can be used to improve security in earlier phases and provides management with information on budget needs

TLDR: Each phase of mobile application penetration testing – analysis, exploitation and reporting – is of equal importance to ensuring a successful test is completed and that remediation actions have been taken.



Making Mobile App Pen Testing Par for the Course

Remember: [One in four](#) mobile apps are hiding a security flaw. If you're a typical organization, you are using between [172 and 255](#) apps on average. That means you are sitting on anywhere between 43 and 64 vulnerabilities in your mobile app stack alone – and if you do nothing, those vulnerabilities will most likely be discovered by attackers. Unless of course, you discover them first.

While supply chain security management is always vital, organizations still need to go the extra proactive mile and approach mobile apps like an attacker – searching for opportunities, vulnerabilities, and ways to leverage weaknesses in real-time. You can't trust your developers 100%; at least 78% of projects have at least one error caused by a developer's misunderstanding of security concepts. And you can't trust most codebases; as software analysis company BlackDuck states, "open source components and libraries form the backbone [96%] of nearly every application in every industry," and upon testing, "seventy-four percent of [risk-assessed] codebases contained high-risk vulnerabilities."

You may not be able to make the mobile app development process safer overnight, but you can largely negate the risks by making Mobile App Pen Testing a part of your regular cybersecurity routine. You wouldn't ingest a vegetable into your system without washing it first. Think of Mobile App Pen Testing as good hygiene for your applications and a necessary precaution before allowing them to interact openly with your environment.

TLDR: With individuals and organizations using numerous apps on a daily basis, your odds of an attack increase exponentially. That said, a mobile application penetration test will significantly reduce risk for your organization and your end users.

FORTRA®

About Fortra

Fortra is a cybersecurity company like no other. We're creating a simpler, stronger future for our customers. Our trusted experts and portfolio of integrated, scalable solutions bring balance and control to organizations around the world. We're the positive changemakers and your relentless ally to provide peace of mind through every step of your cybersecurity journey. Learn more at fortra.com.

