

Mobile Application Security Assessment

PIA VPN

08/19/2025 - Product Version 4.0.21

SCOPE VERIFIED:

PIA VPN

DATE OF COMPLETION

August 19, 2025



PIA VPN

Test Results

All the requirements were met.

Pass

PACKAGE NAME	com.privateinternetaccess.android
TITLE	PIA VPN
DEVELOPER	PIA Private Internet Access, Inc
SHA-256 HASH	f9d2f77f70b771e87508dc6d5a3d7f3e97a317ec6029e4fac88ba923752a6201
SIZE	51.01MB
VERSION CODE	685
VERSION NAME	4.0.21
DEVICE	Google Pixel 7
API LEVEL	35
CERTIFICATION LEVEL	Level 2

Test Background

The Open Web Application Security Project (OWASP) has been around for over 20 years and has helped provide a much more secure experience for both web and mobile users. More recently, it published the Mobile Application Security Verification Standard (MASVS), which aims to define a common standard for secure mobile applications. With the App Defense Alliance, Google has brought together application developers and independent security labs in an effort to improve the security of mobile application security and highlight those apps that meet the standard. The security labs verify the applications against specific MASVS requirements and work with developers to address any issues.

OWASP also publishes the Mobile Security Testing Guide (MSTG), which details how the application should be tested, and provides information to developers on how to write more secure applications. The following section is taken directly from the MSTG to highlight current security best practices, as well as link to additional resources for application developers.

The scope of this work was limited to the specific requirements of the Application Defense Alliance described below and should not be read as a holistic security evaluation or comprehensive penetration test.

Passed Requirements

CATEGORY	STATUS
AUTH-1: If the app provides users access to a remote service, some form of authentication, such as username/password authentication, is performed at the remote endpoint	Pass
AUTH-2: If stateful session management is used, the remote endpoint uses randomly generated session identifiers to authenticate client requests without sending the user's credentials	Pass
AUTH-3: If stateless token-based authentication is used, the server provides a token that has been signed using a secure algorithm	Pass
AUTH-4: The remote endpoint terminates the existing session when the user logs out	Pass
AUTH-5: A password policy exists and is enforced at the remote endpoint	Pass
AUTH-6: The remote endpoint implements a mechanism to protect against the submission of credentials an excessive number of times	Pass
AUTH-7: Sessions are invalidated at the remote endpoint after a predefined period of inactivity and access tokens expire	Pass
CODE-1: The app is signed and provisioned with a valid certificate, of which the private key is properly protected	Pass
CODE-2: The app has been built in release mode, with settings appropriate for a release build (e.g.non-debuggable)	Pass
CODE-3: Debugging symbols have been removed from native binaries	Pass
CODE-4: Debugging code and developer assistance code (e.g. test code, backdoors, hidden settings) have been removed. The app does not log verbose errors or debugging messages	Pass
CODE-5: All third party components used by the mobile app, such as libraries and frameworks, are identified, and checked for known vulnerabilities	Pass
CODE-9: Free security features offered by the toolchain, such as byte-code minification, stack protection, PIE support and automatic reference counting, are activated	Pass

CRYPTPO-1: The app does not rely on symmetric cryptography with hardcoded keys as a sole method of encryption

Pass

CRYPTO-2: The app uses proven implementations of cryptographic primitives

Pass

CRYPTO-3: The app uses cryptographic primitives that are appropriate for the particular use-case, configured with parameters that adhere to industry best practices

Pass

CRYPTO-4: The app does not use cryptographic protocols or algorithms that are widely considered deprecated for security purposes

Pass

CRYPTO-5: The app doesn't re-use the same cryptographic key for multiple purposes

Pass

CRYPTO-6: All random values are generated using a sufficiently secure random number generator

Pass

NETWORK-1: Data is encrypted on the network using TLS. The secure channel is used consistently throughout the app

Pass

NETWORK-2: The TLS settings are in line with current best practices, or as close as possible if the mobile operating system does not support the recommended standards

Pass

NETWORK-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

Pass

PLATFORM-1: The app only requests the minimum set of permissions necessary

Pass

PLATFORM-2: All inputs from external sources and the user are validated and if necessary sanitized. This includes data received via the UI, IPC mechanisms such as intents, custom URLs, and network sources

Pass

PLATFORM-3: The app does not export sensitive functionality via custom URL schemes, unless these mechanisms are properly protected

Pass

PLATFORM-4: The app does not export sensitive functionality through IPC facilities, unless these mechanisms are properly protected

Pass

STORAGE-1: System credential storage facilities need to be used to store sensitive data, such as PII, user credentials or cryptographic keys

Pass

STORAGE-2: No sensitive data should be stored outside of the app container or system credential storage facilities

Pass

STORAGE-3: No sensitive data is written to application logs

Pass

STORAGE-5: The keyboard cache is disabled on text inputs that process sensitive data

Pass

STORAGE-7: No sensitive data, such as passwords or pins, is exposed through the user interface

Pass

STORAGE-12: The app educates the user about the types of personally identifiable information processed, as well as security best practices the user should follow in using the app

Pass