PENETRATION TESTING OF NATIVE APPLICATIONS

Many businesses use native applications to support their internal and business processes. These applications are typically run in a client environment and connect to an application server or, in some cases, directly to a database server.

The security of native applications should not be underestimated, as a successful vulnerability exploitation can lead to the compromising of a client station and an associated data leakage. API service endpoints, where the application connects, may also be vulnerable.

Our team of professional testers is ready to help businesses secure apps on various platforms.

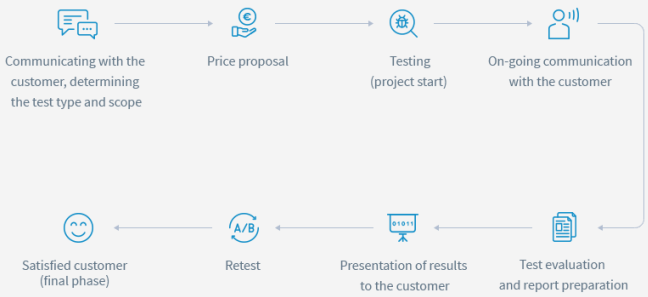
GENERAL SERVICE OVERVIEW

The purpose of the native application penetration test is to detect vulnerabilities in applications and their communication interfaces, demonstrate the exploitation of identified vulnerabilities, identify their risk, and recommend solutions to eliminate detected vulnerabilities.

The penetration test results in a report that does not contain any "false-positive" findings, but only verified vulnerabilities.

The penetration test can be performed using the BLACK BOX (the application source code is not available), GRAY BOX (partial information about the tested app), or WHITE BOX (the source codes of the application are provided at the start of the testing).

The cost of the penetration test depends on the size and complexity of the tested application. This price is determined after customer consultation, which determines the scope, type of test, and other requirements from the customer.



DETAILED OVERVIEW OF THE SERVICE AND METHODOLOGY

Due to their unique nature, native applications require a specific approach. Many of them use proprietary communication protocols and have a unique design, therefore the automated scanners are hardly effective.

Testing includes a range of tests aimed at a local application, its behaviour in the system, server communication, and verification of services on the server (REST API, SOAP, proprietary protocols, etc.).

A comprehensive verification of native application security consists of static and dynamic analysis.

Static analysis is mainly performed through source code review or reverse engineering, followed by manual or automated discovery for vulnerabilities.

Dynamic analysis is performed by debugging on a running system, with testers focusing on potential client issues as well as on server-side issues. Dynamic testing also involves looking for vulnerabilities using various fuzzing techniques.

Binary House combines static and dynamic analysis techniques to improve an efficiency of vulnerability discovery and verification.