141 W. Jackson Blvd, Suite 2100
Chicago, IL 60604
Tel: +1 312-878-1100

September 30, 2025

Dear Sir or Madam:

NowSecure offers app certification for applications which pass our rigorous mobile app security assessment. The assessment covers the entire mobile attack surface including forensic, network and advanced security tests. The certification enables our clients to promote user trust and encourage secure application design and verification in the general marketplace.

NowSecure has performed security testing on Cape's mobile application (co.cape.matrix) version main-f4cb474e-base for Android, and (co.cape.matrix) v1.0 for iOS, with the most recent assessment performed in September 2025. The application was assessed per the details outlined in the "Addendum: Scope of Work" document below. The app was found through empirical testing to avoid significant security vulnerabilities, does not store or transmit sensitive data insecurely, and has effectively implemented various detection mechanisms which help prevent various advanced attack scenarios.

Having passed the assessment, the Cape application for iOS and Android has been issued NowSecure Certification for the platforms and versions tested.

Please don't hesitate to reach out to me with any questions.

Sincerely,

Katie Strzempka
SVP Customer Success and Services

NowSecure™

141 W. Jackson Blvd, Suite 2100
Chicago, IL 60604
Tel: +1 312-878-1100

Addendum: Scope of Work

NowSecure Process

Our process provides a much more thorough assessment than static code review can provide.

NowSecure utilizes several approaches to fully test your mobile applications. An important distinction is that our testing is performed by qualified **forensic and security analysts**, not simply through automated code analysis. While we have automated some processes in order to provide a cost-effective solution, much of our assessment requires cutting-edge testing not available in code analysis tools.

Our assessment includes both authenticated and non-authenticated testing which accurately models the security challenges app providers face. For non-authenticated testing, we mimic the types of analysis and attacks used by cyber criminals. These include reverse engineering the apps, static and dynamic analysis, DNS and Web attacks, man-in-the-middle and more.

Furthermore, our skilled forensic investigation uncovers artifacts which may be recoverable after a user has authenticated and utilized the application. These include insecure storage of sensitive data, circumventing passcodes once logged in, 2-factor authentication, and keychain artifacts.

App testing criteria

To help illustrate our comprehensive testing, the following list is provided for your reference. This list may not be exhaustive as our continual research and development add new testing criteria on a regular basis, ensuring our tests remain up-to-date with the latest exploits and security issues.

1. Avoid insecure storing of user name
2. Avoid insecure storing of password
3. Avoid insecure storing of application data on device
4. Avoid storing of application images or screen captures
5. Implement 2-factor authentication correctly
6. Avoid insecure storing of any 2-factor authentication data
7. Avoid risk of storing sensitive user data in a backup
8. Avoid insecurely transmitting login data
9. Avoid vulnerability to man-in-the-middle (MITM) SSL attacks
10. Avoid insecurely transmitting sensitive user data over Wi-Fi
11. Avoid risk for session hijacking
12. Avoid Storing sensitive data in history artifacts
13. Enable function to permanently delete user data
14. Securely handle interruptions to transactions
15. Avoid insecure storing of data on removable media such as SD
16. Platform security risks

NowSecure™

141 W. Jackson Blvd, Suite 2100
Chicago, IL 60604
Tel: +1 312-878-1100

17. Enforce SSL-only
18. Secure against web penetration such as framing, phishing, XSS
19. Avoid data loss in log files or application debugging information
20. Encryption properly implemented and sufficient for data protection
21. Code secured against reverse engineering
22. Dynamic code injection at runtime
23. API fuzzing as needed
24. Device exploitation as needed
25. Avoid device/app cloning

Additional security issues as discovered