COSSACK LABS

Xumm Mobile Wallet Security Review: Executive Summary

Overview and executive summary

In August 2021, XRPL Labs requested Cossack Labs offer an opinion on improving the security and cryptography aspects of Xumm mobile wallet's behaviour, source code, and cryptographic design.

Xumm is a hot, self-custodial mobile cryptocurrency wallet for the XRPL blockchain ecosystem that provides users with the ability to manage their private keys, make payments and interact with the XRPL via xApps.

Goals and risk statement

Review goal: Improve security and cryptographic aspects of Xumm and the surrounding ecosystem, relevant to the security of the private keys in the self-custodial wallet itself and the transactions it signs.

Risk statement:
XRPL Labs makes the public security claim that Xumm is "built with security as a priority" and that the Xumm application behaves securely as long as users follow the following recommendations:

- *Users are keeping their account secret (family seed/mnemonic/Secret Numbers) safe*
- *Users are keeping their personal phones up-to-date and protected*
- *Users follow modern mobile security practices*

Thus, security components inside Xumm wallet should be sound against the following risk statement (C.A.S.E. model):

> *Financial loss (consequence) due to active and passive adversaries (source) exploiting application security and cryptography flaws (event) in Xumm mobile app, resulting in unauthorised usage of secrets, keys and other identifying material (assets) to perform unauthorised transactions.*

The risk statement above is limited to the Xumm application itself. The security of the mobile device, mobile OS, and other installed applications is out of direct influence and scope and lies on the user. As XRPL Labs stated in a public security claim, "You should make sure your device (smartphone) is up to date with OS security updates and patches".

Xumm mobile wallet security review: executive summary

For: XRPL Labs

Public, shared on demand. 18.05.2023

COSSACK LABS

In scope: Private key life cycle (seed creation, seed storage, seed import/export), transactions correctness (transactions falsification and abuse), cryptography design and implementations around key lifecycle and transactions, preventing unauthorised access to the wallet, preventing adversarial inputs or tricking the users into signing unauthorised transactions; secure management of sensitive data, usage of 3rd party libraries and components, data leakage from the wallet application, application <> backends communication (relevant to the threat modelling).

Non-scope for risk statement:

- The functionality and safety of XRPL mainnet is out of scope of the current review.
- It is understood that Xumm wallet security controls might be partially or fully compromised if the mobile device is jailbroken/rooted, or is under a Pegasus-like attack. Protections against mobile device compromise (jailbreak or root) are out of scope.
- It is understood that anyone with access to a mobile device has partial or full control over Xumm wallet.
- Security and safety of 3rd party backends and xApps that Xumm wallet communicates with, is out of scope.
- 3rd party implementations of cryptographic primitives are considered trusted or can be replaced by well-studied ones.

Coverage:

The initial security review has been based on:

- Code: commit 4ebb0861bbd0355218b6fa64dfd4042862c9589a https://github.com/XRPL-Labs/Xumm-App/commit/4ebb0861bbd0355218b6fa64dfd4042862c9589a
- Design documentation and components description provided by the XRPL Labs team.

The initial review was conducted using the following devices:

- iPhone 12 Pro (iOS 15.0.2)
- iPhone XS (iOS 14.7.1)
- iPhone 11 Pro (iOS 14.6)
- iPhone 7 (iOS 13.1.3)
- Xiaomi Redmi Note 9 (Android 10)
- AVD (Android 10)
- AVD (Android 7.1.1)

Methodology

The review included the following activities:

- **General risk model clarification and security review**: Formulating realistic risk models and threat vectors that affect user safety.
- **Research of fundamental issues of cryptocurrencies in context of interacting with wallet**: Analysing common issues inherent to blockchain/network and linking them to particular wallet's issues.
- **Design/architecture review**: Proactively seeking design flaws that lead to leakage of sensitive data stored within a Xumm app or manipulating the transaction flow.
- **Cryptographic design and implementation review**: Verifying whether the chosen combination of cryptographic primitives and their implementation actually embodies desired security properties.
- **Application security**: Ensuring that application-level security controls are implemented well, mitigating platform-specific threats and supply chain risks.

Applicable standards

During the assessment, our work was driven by industry experience and (where applicable to a reasonable extent) the following standards:

Review baseline: OWASP MASVS v1.5 L1+L2 (Mobile application security verification standard), OWASP ASVS v4.0.3 (Application security verification standard).

Industrial standards and recommendations: NIST SP 800-57 (Recommendation for Key Management), NIST SP 800-53 (Security and Privacy Controls for Federal Information Systems and Organizations), NIST SP 800-37 (Risk Management Framework).

Applicable platform standards: Apple platform security guidelines, Android security best practices, React Native security best practices.

Triaging issues

Due to the specific risk statement/review goal, issues discovered could not be triaged using a common methodology like CWE or CVSS – the outcomes, loss magnitude in the general context are significantly different compared to a regular security assessment and some of the vulnerability severity scores would be misleading.

Issues are triaged as critical, high, medium, or low based on a performed risk assessment and a formulated trust and risk model representing the chosen risk statement.

Findings summary

During the initial review, we found zero critical issues, few high-priority bugs, risky design decisions, and recommended the fixes. Also, security improvements and “raising the bar” security controls have been advised.

Because Xumm is a self-custodial application, OWASP MASVS v1.5 was used as a baseline for security assessment, excluding requirements that were out of scope (such as V8 Resilience Requirements), and including some relevant requirements from OWASP ASVS v4.0.2 (such as security HTTP headers during network communication).

The resulting list of security findings and recommendations can be found below:

Findings area	Critical	High	Medium	Low
Software design	0	0	6	3
Cryptography (design & implementation)	0	5	3	3
Application security	0	2	3	8
Platform security	0	1	1	4
Code quality	0	0	1	3
Infrastructure	0	0	0	3
Supply chain risks	0	0	1	0

Overall findings: 47 findings (0 critical, 8 high, 15 medium, 24 low).
OWASP MASVS coverage: 43% (28 satisfied requirements out of relevant 65).

This initial review was performed during September-October 2021. It consisted of around 240 person-hours of work, allocated between design review, risk/threat modelling, implementation review, cryptographic review and verification of the chosen security controls.

Joint activities on re-designing and re-implementing the cryptographic layer were performed during August-October 2022.

Verification of fixed issues

Verification of implemented fixes was performed during January-April 2023 and showed a significantly decreased amount of the remaining issues:

Findings area	Critical	High	Medium	Low
Software design	0	0	3	0
Cryptography (design & implementation)	0	0	1	0
Application security	0	0	0	3
Platform security	0	0	0	1
Code quality	0	0	0	3
Infrastructure	0	0	0	0
Supply chain risks	0	0	0	0

Findings left: 11 findings (0 critical, 0 high, 4 medium, 7 low),
OWASP MASVS coverage: 89% (58 satisfied requirements out of relevant 65).

After XRPL Labs team has fixed the acknowledged issues, and the Cossack Labs team has verified
the fixes, the current status is the following:
- “Fixed”, or “Partially fixed” – 32 findings (0 critical, 7 high, 10 medium, 15 low);
- “Scheduled, not fixed” – 11 finding (0 critical, 0 high, 4 medium, 7 low);
- “Acknowledged but won’t do” – 4 findings (0 critical, 1 high, 1 medium, 2 low).

See Findings details to learn the status of each finding.

Full report

The full report package consists of four separate files:

- The initial executive summary that provides a high-level overview of the performed audit and found issues.
- An updated version of the executive summary (this document), produced after verification of the fixed issues.
- A technical report which describes the Xumm architecture, risk model, threat model, already implemented security controls, cryptography primitives inventory, cryptography operations review, application security review, code quality, and review of supply chain risks.

- Appendix A provides technical details of each finding, its fix, and verification status.

Conclusion

During the security review, we evaluated the implementation of Xumm mobile application, its architecture, theoretical and practical concerns. We reviewed Xumm app secure storage, its interaction with users, and communication with the XRPL network, Xumm backend and xApps.

Our impressions after the initial audit

Xumm covered **43%** of relevant security requirements from OWASP MASVS: 28 out of 65. This is a single score for both iOS and Android applications. Overall, 47 issues were found, including some broken/missing security controls that, in the event of an unfavourable circumstance, such as unauthorised access to a mobile phone, could result in the leakage of users' sensitive data, the loss of account data, or the initiation of unauthorised transactions. No critical vulnerabilities or immediate exploits were identified.

Our general conclusion is that the set of security controls - implemented by the XRPL Labs team - can achieve security claims and prevent risk statements to a satisfactory degree because Xumm app meets the majority of OWASP MASVS Level 1 security requirements (known as “basic security requirements”).

We have found out that if Xumm users understand:

- their responsibilities of protecting their account credentials,
- limitations of mobile applications and hot self-custodial wallets,
- their responsibility of ensuring mobile device and mobile OS security prior to unsealing the wallet,

… then users’ data, keys, and transactions are acceptably protected with Xumm’s security measures.

We would like to note a solid security-oriented engineering effort of the XRPL Labs team in building and securing the app. They put in a lot of effort and implemented many security controls, like application locking, protecting user accounts with a passphrase, data-at-rest encryption, user confirmation on sensitive actions, filtering xApps, filtering network requests, and many more.

As the main goal of this engagement was to improve the security of Xumm app, we also separately provided suggestions on application security, cryptography, design mitigating platform-specific risks, improving general stability and maintainability of Xumm wallet by building defense-in-depth protections.

Such protections could significantly “raise the bar” for attackers, decreasing the chances of incidents that are out-of-direct control by application developers, like mobile OS exploits.

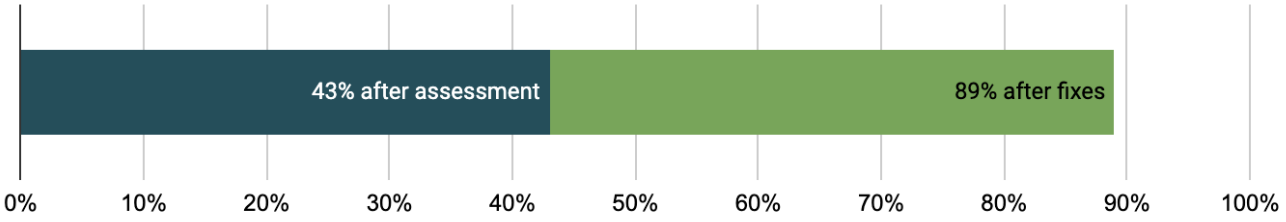
Our impressions after the verification of fixed issues

After reviewing all of the reported findings, the XRPL labs team prioritised fixing high and medium-severity issues first. The XRPL Labs team resolved all high-level issues, which were then verified and confirmed by the Cossack Labs team. As the Xumm app development continues, 11 of the 47 findings are scheduled for future releases, and are listed as "Scheduled, not fixed”.

The security score increased to **89%**: now Xumm app covers 58 out of 65 security requirements based on OWASP MASVS v1.5.

While the Cossack Labs team verified the fixed issues, there was no complete re-evaluation of all the changes in the source code that were made over time.

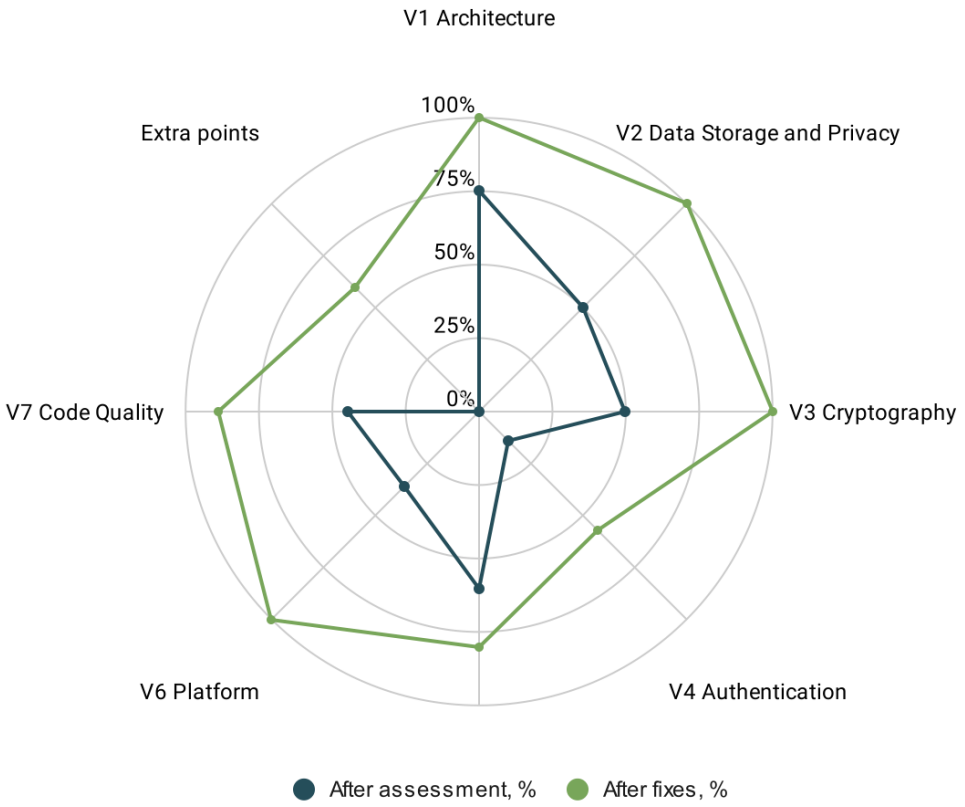
Taking into consideration the feedback, the XRPL Labs team re-implemented the entire cryptographic layer responsible for application-level encryption and sensitive data storage, introducing the "v2" encryption scheme. Changing the encryption layer resolved the majority of the cryptography section's issues and weaknesses.



Xumm satisfied 43% of security requirements after the assessment, and 89% after verification of fixes.

We see significant improvements in handling application security corner cases, managing third-party dependencies, and applying platform-specific security controls and settings.

Many "medium" and "low" issues have been resolved by the Xumm team. These security controls contribute to building defense-in-depth: a solid foundation for future application development and mitigation of currently unknown threats.



Security requirements covered by Xumm app based on OWASP MASVS v1.5. Each chapter represents a certain application security topic and has a different number of requirements, calculated as 100%.

We see that XRPL Labs is dedicated to continuously improving the security of their product and they recognize that there are still some areas for enhancements. Many of our recommendations are scheduled to be addressed in future releases.

Findings

The findings are triaged into categories depending on the area, type of issue, and severity of the outcome.

Types: Broken controls (security controls that are implemented but don't satisfy security requirements), missing controls (lack of protections), improvements (security-related suggestions to implement).

Areas:

- Design (D) – security design and architecture issues.
- Crypto (C) — cryptography-related issues.
- Application security (A) – issues related to mobile application security.

- Platform (P) – mobile platform (devices and OS) issues.
- Code quality (CQ) — typos, documentation, and general code style, testing.
- Infrastructure (INF) – backend configurations, TLS settings, CICD pipelines.
- Supply chain management (S) – management of third-party dependencies.

The statuses of issues could be the following:

- **Fixed** – the issue was confirmed by the XRPL Labs team, the fix was implemented in full according to the platform limitations, and validated by Cossack Labs team.
- **Partially fixed** – the issue was confirmed by the XRPL Labs team, the fix was partially implemented and validated by Cossack Labs team.
- **Scheduled, not fixed** – the issue is confirmed by the XRPL Labs team, planned and scheduled for later releases, currently not fixed.
- **Won't do** – the issue was confirmed by the XRPL Labs team, the associated risks were accepted, and/or compensating security controls were implemented.

Findings and references

ID	Problem	Severity/ priority	Type	Area	Status
D-001	Require passphrase to delete an account's private key	Med	missing	design	Fixed
D-002	Tie biometry check to Keychain/Keystore	Med	impr	design	Won't do
D-003	Very weak passcodes are allowed	Med	impr	design	Fixed
D-004	Lack of brute-force countermeasures for account passphrase	Med	missing	design	Scheduled, not fixed
D-005	Improve the account passphrase's rules	Med	impr	design	Scheduled, not fixed
D-006	Imported mnemonic does not follow bip39	Med	impr	design	Scheduled, not fixed
D-007	Account secret/Family seed is exposed through UI	Low	missing	design	Fixed
D-008	Require device-level passcode	Low	missing	design	Partially fixed
D-009	Clear Keychain on app reinstall	Low	missing	design	Fixed
C-001	Android: possible use of AES-CBC encryption with null IV	High	broken	crypto	Fixed

C-002	Accounts private keys are encrypted with AES-CBC — use AEAD instead	High	impr	crypto	Fixed
C-003	iOS: possible use of AES-CBC encryption with nil IV	High	broken	crypto	Fixed
C-004	Use password-based KDF to derive encryption key from app passcode and account passphrase instead of sha256	High	broken	crypto	Fixed
C-005	Improve application passcode's hashing	High	broken	crypto	Fixed
C-006	Realm encryption key is never explicitly removed	Med	missing	crypto	Fixed
C-007	Passcode is used for authentication and encryption – improve its usage and storage	Med	impr	crypto	Fixed
C-008	Improve questionable design of bearer token	Med	impr	crypto	Scheduled, not fixed
C-009	Clear sensitive data memory after use	Low	missing	crypto	Partially fixed
C-010	Untie encryption keys from application passcode / account password	Low	impr	crypto	Won't do
C-011	Minimize a lifetime of Realms' encryption key	Low	impr	crypto	Fixed
A-001	iOS Biometry check bypass is possible: track biometry change	High	broken	appsec	Fixed
A-002	Disable JavaScript in WebView	High	broken	appsec	Won't do
A-003	Improve weak settings of react-native-keychain to store account's private key	Med	broken	appsec	Fixed
A-004	Clear WebView cache	Med	broken	appsec	Fixed
A-005	xAppBrowser WebView opens unsanitized links	Med	impr	appsec	Fixed
A-006	Lack of certificate pinning	Low	missing	appsec	Scheduled, not fixed
A-007	Improve the Keychain settings	Low	impr	appsec	Won't do
A-008	HTTP traffic allowed for localhost	Low	impr	appsec	Fixed
A-009	iOS app screen is not blurred when moved to the background	Low	broken	appsec	Fixed

A-010	WebViews should allow only the minimum set of protocol handlers	Low	impr	appsec	Fixed
A-011	Counter of failed passcode attempts not always shows real time left	Low	broken	appsec	Scheduled, not fixed
A-012	Clean up sensitive data in case of security event (brute force, reset)	Low	impr	appsec	Fixed
A-013	Improve access token refresh and expiration	Low	impr	appsec	Scheduled, not fixed
P-001	Prevent Android overlay attacks	High	missing	platform	Fixed
P-002	Enable out-of-the-box Android security controls	Med	missing	platform	Fixed
P-003	Enable Exported flag for activities/services/receivers with intent filters	Low	broken	platform	Fixed
P-004	Remove Debug code from the Release build	Low	impr	platform	Fixed
P-005	Analyse and remove unnecessary Android permissions	Low	impr	platform	Fixed
P-006	Users could be tricked by xApp Webview	Low	broken	platform	Scheduled, not fixed
CQ-001	Improve misleading comments and names in crypto security controls	Med	impr	code	Fixed
CQ-002	Code quality: Android analyser errors	Low	impr	code	Scheduled, not fixed
CQ-003	Code quality: Xcode static analyser errors	Low	impr	code	Scheduled, not fixed
CQ-004	Code quality: low priority issues identified by SASTs	Low	missing	code	Scheduled, not fixed
INF-001	Improve CSP directives	Low	impr	infra	Fixed
INF-002	Remove redundant headers on the back-end	Low	impr	infra	Fixed
INF-003	Lack of rate-limiting for requests on the Xumm wallet back-end	Low	missing	infra	Fixed
S-001	Third-party dependencies policy requires revision	Med	broken	supply chain	Partially fixed

About Cossack Labs

Cossack Labs is a provider of data security tools (cryptographic and data security frameworks), bespoke solutions and consulting services, with a focus on sensitive data protection in modern systems. Cossack Labs' experts participating in this audit, have decades of hands-on practical experience, appropriate formal education and academic degrees in cryptography, software engineering, data security and general information security. Cossack Labs' security engineers are acknowledged contributors to popular industry standards (OWASP MAS) and hold appropriate certifications (CISSP).

Due to the nature of our skillset, our review aims not only to detect potential weaknesses but also to provide clear, actionable advice for developers to rapidly improve security in their applications, as communicated by thinking-alike engineers.

Cossack Labs can be contacted at: cossacklabs.com / info@cossacklabs.com

0.1	28 October 2021	Cossack Labs team	Initial version of the executive summary, technical report and appendix with technical issues.
0.2	18 May 2023	Cossack Labs team	Updated executive summary after the verification of fixes: added statuses of each issue, added section "Verification of fixed issues", added section "Our impressions after the verification of fixed issues".