Quick Security Evaluation

Perspectives

A Security Design Evaluation

V 1.0
Amsterdam, October 19th, 2023

Contact

For more information about this document and its contents please contact Radically Open Security B.V.

Name	Melanie Rieback
Address	Science Park 608 1098 XH Amsterdam The Netherlands
Phone	+31 (0)20 2621 255
Email	info@radicallyopensecurity.com

1. Introduction

Between 26-09-2023 and 05-10-2023, Radically Open Security B.V. carried out a Security Evaluation for Perspectives. It was a targeted engagement designed to provide both guidance and validation for the security enhancements that were planned in response to findings from a previous security assessment. A total of five workdays were invested to ensure a comprehensive review and expert advice on the planned measures.

The scope of this security evaluation encompassed both the client-side application, which is accessible by the url https://mycontexts.com/, and the server-side supporting services found on this url; namely the RabbitMQ messagebus.

In this report, the primary objective is to reevaluate the decision to adopt Decentralized Identities (DID's) within our current application architecture. The central question we aim to answer is whether DIDs offer an efficient solution to previously identified security concerns.

Should DIDs remain the chosen direction for our security framework, the report will then shift its focus to the application and implementation of these Decentralized Identities. Particular attention will be given to the functionalities that have been selected and their implications for the application's security, both positive and negative.

Ultimately, this document will serve as a comprehensive evaluation report, outlining and justifying the selected security approaches.

2. Security Evaluation for Perspectives

Tasks Performed

- Collaborated with Project Representative through messaging and video conferencing to tailor an advisory that aligns with the prospective technical design of new functionalities in Perspectives.
- Conducted a functional analysis of Mycontexts.com.
- Performed a technical analysis of the current state of Mycontexts.com.
- Reviewed recommendations provided in the report 'report_ngid-perspectives2022' from one year ago.
- Analyzed version 5 of the document 'A Security Perspective on the Distributed Runtime'.
- Conducted a comprehensive analysis of the technical design document 'Identities and Key Management', including its planned design choice for Decentralized Identities (DIDs) and the associated rationale.
- Studied possible alternative solutions to the problem of Globally Unique Identifiers and collisions.
- Discussed and reviewed sharing public keys in a way that authenticity is guaranteed.
- Reviewed design choices regarding the safe storage of the private keys of a user.
- Penetration tests have been performed on the assets, targeting both server-side and client-side potential vulnerabilities.
- In a workshop-like setting, a robust method of threat modeling was collaboratively discussed with the client. The model was then provided and examined, serving as a foundation upon which the representative can continue to build.

Security Considerations

- Opt for the simplest design that still ensures robust security: Simplicity often reduces the number of potential vulnerabilities and makes the system easier to audit.
- Avoid temporary insecure choices made for clarity or rapid development: These can pose significant security risks if not rectified before deployment.
- Remain vigilant for vulnerabilities that may easily creep in due to ongoing product development and the passage of time.

Recommendations

- Reconsider the choice of using DIDs due to the complexity they introduce. An alternative could be Cuid2, designed to virtually eliminate the risk of collisions.

- Maintain a record of all essential TODOs to be addressed before deployment: This ensures that no crucial steps, especially related to security, are overlooked. The client has proactively compiled a list of TODOs, based on both our verbal discussions and their own expertise. This list includes items like addressing plain text passwords in RabbitMQ connections, securing password information in fields, and encrypting the contents of Deltas, among other client-identified items. The list has been shared on this url: https://mycontexts.com/techdoc/_security_to_do_list.html for easy access and ongoing updates.
- Ensure robust version control to promptly address newly discovered Common Vulnerabilities and Exposures (CVEs) in third-party libraries and software.

The Challenges with GUIDs

As highlighted in our 2022 report, the presumption that GUIDs (Globally Unique Identifiers) ensure uniqueness is unfortunately not foolproof.

The requirement for unique identification within Perspectives is of paramount importance. Despite the assurances implied by their name, GUIDs have been found to be prone to collisions. Such collisions can result in significant functional issues, jeopardizing data integrity and negatively impacting user experience. Given these findings, it is imperative for organizations to re-evaluate the reliability of GUIDs, particularly when employed as primary context keys in distributed systems.

Cuid2 as an alternative to DIDs

Cuid2 aims to address the shortcomings of GUIDs by offering better collision resistance and security. It combines multiple sources of entropy in a one-way hash function to produce truly unique IDs. It's lightweight, horizontally scalable, and compatible with offline operations.

DIDs offer another solution for generating unique IDs in a decentralized manner, often utilizing blockchain technology. While they are robust and secure, they are also more complex to implement and may require an internet connection for validation.

Complexity:
Cuid2 is simpler and easier to implement, with a small code footprint. DIDs, while powerful, bring in additional complexity due to their dependence on distributed networks or blockchains.

Conclusion: It is advisable to implement CUID2 instead of DIDs as a solution for the risk of collisions, due to its reduced complexity while maintaining strong security.

Other potential solutions for Unique Identification

Several alternatives have been considered to enhance the system's reliability by using improved Globally Unique Identifiers. These alternatives are UUIDv5, and ULID.

UUIDv5: Version 5 of the Universally Unique Identifier scheme employs a deterministic method to generate IDs based on name and namespace. It's a robust and widely-accepted standard, but may not be well-suited for systems that require dynamic, non-name-based IDs.

ULID: Universally Unique Lexicographically Sortable Identifier is designed to offer uniqueness like UUIDs, but also adds a time component that allows for sorting. It's a suitable option for systems where the sortability of identifiers is beneficial.

Upon reviewing the specifications, it appears that these two alternatives may not align as well with the nature of Perspectives as the previously mentioned CUID2 does

Public Key Sharing and Sender Authenticity

In the original design, Decentralized Identities (DIDs) served the purpose of ensuring the authenticity of the sender when sharing a public key from a public-private key pair. An alternative approach, suggested by the representative of Perspectives, is to include the public key in the employee invitation procedure. This occurs via an invitation file, and peers are expected to exchange these files securely, independent of Perspectives.

Advice

- Utilizing the invitation file for sharing the public key is generally a secure method. However, it is imperative that users are made aware that by accepting and importing the invitation file, they are effectively confirming the authenticity of the peer. Users must be cognizant of this responsibility and perform due diligence to verify authenticity.
- Additionally, Perspectives's planned design includes a future handshake procedure aimed at adding an extra layer of verification. While the details are yet to be fully defined, a verbal verification, possibly via phone, would be an additional security measure.

Signing and authenticating deltas

The design-document contains a chapter about Signing and authenticating deltas. The author describes the planned use of the SubtleCrypto interface of the Web.Crypto.API. This design-choice is carefully reviewed.

Advice

SubtleCrypto is widely recognized as a secure method for basic cryptographic functions such as hashing, encryption, and digital signatures. Setting the extractable property to false is a good practice to make the key non-exportable. Given these considerations, it's a strong choice for cryptographic operations. However, an external review by an

expert in cryptography and security is highly recommended after implementation to identify subtle errors and potential vulnerabilities.

Storing the Cryptographic Key

The design-document contains a chapter about storing the cryptographic key. The author outlines the plan to store the CryptoKey object in IndexedDB. This design-choice is carefully reviewed.

Advice

Storing the CryptoKey object in IndexedDB is in line with the chosen design philosophy that a user's data is as secure as their own device. While additional security layers for the IndexedDB may not be as critical in this context, basic access controls should still be in place. Given the inherent risks, an external review by an expert in cryptography and web security is recommended to validate this approach after implementation and identify any overlooked (primarily client-side) vulnerabilities.

Discovered Vulnerabilities

Upon conducting reconnaissance and penetration testing on the application, the following vulnerabilities have been uncovered:

- Apache httpd server, Version 2.4.52. Initially flagged as outdated, but upon further investigation, it appears to be receiving security patches via Ubuntu's unattended-upgrades feature. While the server version is not the latest, it is still maintained and should be considered secure as long as unattended-upgrades is active and `apt update` and `apt upgrade` are executed regularly. Failure to manually update could result in exposure to vulnerabilities in packages that are not covered by unattended-upgrades.
- At the URL /index.js, a development instance of the webpack_module is active.
- Email addresses of the application's creators or representatives are exposed in the source code. Notably, corbaars@perspect.it and joopringelberg@perspect.it were identified.
- Security issues were observed in the DBCouch instance. URL requests to /cw_servers_and_repositories/_all_docs yielded sensitive information. Similar issues were observed with /cw_perspectives_domains/perspectives_domains-BodiesWithAccounts and /cw_perspectives_domains/_all_docs.
- The Connect-request for the messagebus contains a plaintext password.
- Brute-forcing passwords for the connection procedure seems plausible. Time constraints within the project scope prevented the development of a specialized test script. Nonetheless, preliminary testing suggests there is no imposed limit on the establishment of new connections.

Threat Modeling Workshop Summary

Conducted a comprehensive threat modeling workshop with the client to further deepen his understanding of application security. While the client had been introduced to Threat Modeling in a previous assessment, the session aimed to provide hands-on experience and clarify how to practically implement threat modeling. The workshop was structured around the Threat Modeling Manifesto's 4-questions framework, values, and principles.

Decomposing the Application

Introduced the client to Threat Dragon, a threat modeling application, and collaboratively examined and adjusted an initial model of his application architecture.

Determine and Rank Threats

Explained the STRIDE methodology to him, focusing on how to identify and prioritize threats. Further solidified the concept by engaging in an online Elevation of Privilege (EoP) Threat Modeling Game.

Countermeasures and Mitigation

Touched upon the third step of the threat modeling process, discussing potential countermeasures and mitigation strategies for identified threats with him.

3. Contact Details

Your pentester for this project:

- Peter de Witte
  Peter has 16 years of experience as a software engineer, the majority of which he has spent as an independent entrepreneur. Recognizing the critical importance of cybersecurity, he shifted his focus and is now mostly working as an ethical hacker.
  Expertise:
  Ethical Hacking of Web Applications: Pentesting, advice, and general consultancy
  Secure Coding: Proficient in C#, JavaScript, ReactJS, Python
  Reverse Engineering and Testing of Android Mobile Apps
  Committed to education and inspiration, he shares his extensive knowledge through hands-on demos, workshops and courses, engaging students, entrepreneurs, and industry professionals alike.

If you have any questions about this advice, please contact us at info@radicallyopensecurity.com

For more information about Radically Open Security and its services please visit our website at www.radicallyopensecurity.com

4. Disclaimer

It is important to understand the limits of ROS's services. ROS does not (and cannot) give guarantees that something is secure. ROS, instead, has an obligation to make reasonable efforts (in Dutch: “*inspanningsverplichting*”) to perform the agreed services.

ROS and "client name" agree to take reasonable measures to maintain the confidentiality of information and any personal data they gain access to in the course of performing the test. Both parties will use the information and data they receive or access only for the purposes outlined in this agreement. ROS warrants that all core-team members, external freelancers, and volunteers it engages to perform the test have signed a non-disclosure agreement (NDA).