Talisman Wallet Extension Assessment

Security Assessment Report

TALISMAN

Prepared for Talisman
January 7, 2022 (version 1.1)

Project Team:

Technical Testing	Bryan C. Geraghty and James Cook
Technical Editing	Tom Steele and Lacey Kasten
Project Management	Molly Vukusich

Table of Contents

Engagement Overview

Assessment Components and Objectives

Talisman recently engaged Atredis Partners ("Atredis") to perform a security assessment of the Talisman Wallet browser extension. Objectives included assessing the wallet implementation to ensure that any weakness in browser API implementation, sensitive data handling, and extension logic were identified for remediation.

Testing was performed from December 14, through December 23, 2021 by Bryan C. Geraghty and James Cook of the Atredis Partners team, with Molly Vukusich providing project management and delivery oversight. For Atredis Partners' assessment methodology, please see Appendix I of this document, and for team biographies, please see Appendix II. Specific testing components and testing tasks are included below.

COMPONENT	ENGAGEMENT TASKS
Talisman Wallet Extension Assessment	

Assessment Targets	• Talisman Wallet Extension
	  • ManifestV2 for Chrome
	  • Typical Substrate wallet functionality: Manage keys
	  • Also provides functionality to sign and send arbitrary extrinsic calls
	  • Approval mechanism for URLs to interact with extension

Assessment Tasks	• Source-Assisted Penetration Testing of the Talisman wallet
	  • Assess browser-API implementation and sensitive data handling
	  • Assess trust boundaries
	  • Assess Talisman client-side SDK implementation
	  • PoC generation and validation of findings

Reporting and Analysis	
Analysis and Deliverables	• Status Reporting and Realtime Communication • Comprehensive Engagement Deliverable • Engagement Outbrief and Remediation Review

The ultimate goal of the assessment was to provide a clear picture of risks, vulnerabilities, and exposures as they relate to accepted security best practices, such as those created by the National Institute of Standards and Technology (NIST), Open Web Application Security Project (OWASP), or the Center for Internet Security (CIS). Augmenting these, Atredis Partners also draws on its extensive experience in secure development and in testing high-criticality applications and advanced exploitation.

Engagement Tasks

Atredis Partners performed the following tasks, at a high level, for in-scope targets during the engagement.

Browser Extension Penetration Testing

For browser extensions, APIs and supporting services, Atredis performed automated and manual application penetration testing of these components, applying generally accepted testing best practices. Atredis Partners performed a targeted assessment for specific, critical application components, including but not limited to:

- Sensitive data storage within browser extension and libraries
- Configuration and usage of the browser extension and implementation-specific settings
- Communications security with supporting application server infrastructure
- Browser security permission weaknesses and related local attack surface

The objective of the browser extension component of the engagement was to develop a clear understanding of the working of the browser extension and libraries, identifying areas where the browser extension may present potential for compromise or other exposure.

Binary and Runtime Analysis

For relevant software targets identified during the course of this engagement, Atredis performed binary and runtime analysis, using debugging and decompilation tools to analyze application flow to aid in software security analysis. Where relevant, purpose-built tools such as fuzzers and customized network clients may have been utilized to aid in vulnerability identification.

Source Code Analysis

Atredis Partners reviewed the in-scope application source code, with an eye for security-relevant software defects. To aid in vulnerability discovery, application components were mapped out and modeled until a thorough understanding of execution flow, code paths, and application design and architecture were obtained. To aid in this process, the assessment team will engage key stakeholders and members of the development team where possible to provide structured walkthroughs and interviews, helping the team rapidly gain an understanding of the application’s design and development lifecycle.

Executive Summary

For this assessment, the Talisman development team provided Atredis with access to the Talisman Wallet source repository and indicated that the `fcba645` commit hash was the version to be assessed. Atredis Partners was able to build version `0.5.0` of the extension from that commit.

Testing was performed by connecting the extension to the `https://polkadot.js.org/apps/` application, which was configured to communicate with a development Polkadot[1] node owned by Atredis. The `https://app.talisman.xyz/` application was also used to assess the site authorization controls of the extension. Once the applications were authorized to communicate with the extension, existing Polkadot accounts were imported into the wallet and new extension-owned accounts were created. Then some of the accounts were connected to the applications in order to test access controls.

Next, Atredis Partners performed exploratory testing to gain familiarity with the features and started reviewing the code to gain an understanding of the extension architecture. Finally, Atredis performed dynamic testing and a full source review of all extension functionality. Special attention was paid to content script injection, bootstrapping, HTTP traffic, API message routing, data storage, and API authentication and logic, and manifest configuration.

During the second week of the assessment, the Talisman team requested that Atredis Partners complete the remainder of the assessment on the latest version, which ended up being version `0.6.1`. The observations made in the conclusions and overview sections below apply to both versions.

Key Conclusions

Ultimately, Atredis found the Talisman Wallet to be well-designed and well-implemented. No exploitable weaknesses were identified during this engagement.

As explained in detail in the Browser Extension Overview section below, Atredis made a number of observations that could lead to a more robust application, but none of them represented an exploitable weakness. To summarize, these were:

- Registered account addresses and Authorized URL configurations are stored in plaintext and can be read by the extension even when it is locked
- Some API operations leak data to the extension pop-up context when the extension is locked, but that data is also directly accessible from the extension pop-up context
- A password change feature is planned but not yet implemented

[1] https://github.com/paritytech/polkadot

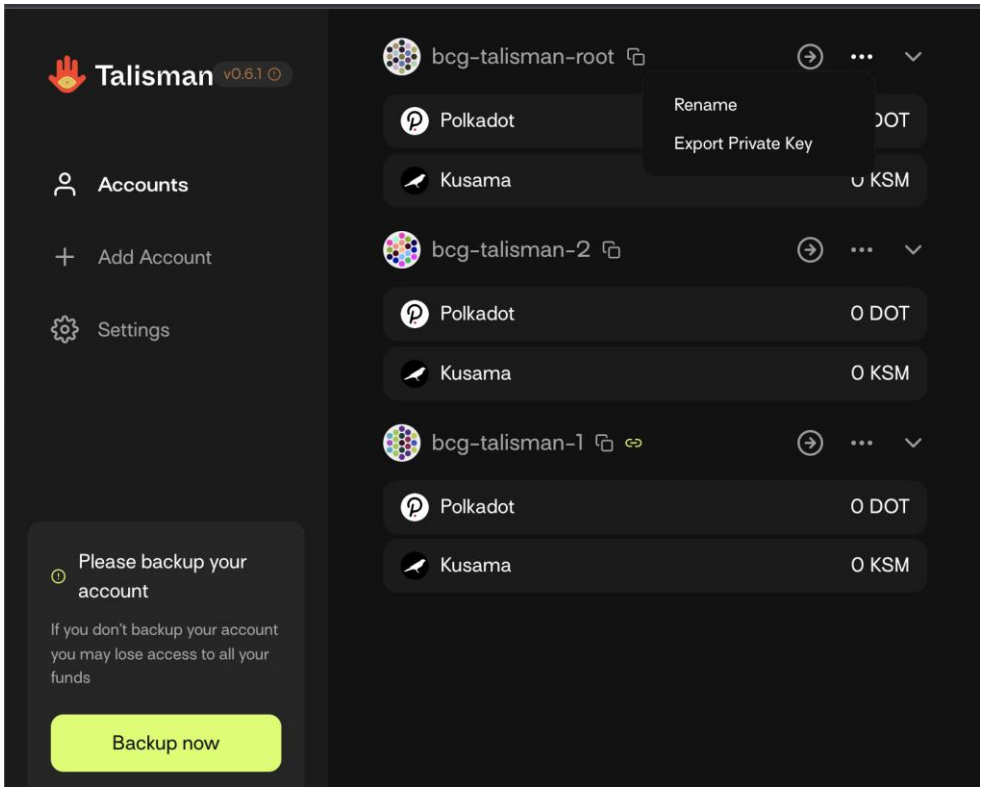
- An auto-lock feature is planned but not yet implemented
- The manifest configuration allows the extension to potentially load into any website, but this is limited by the site authorization controls
- The manifest configuration allows the extension to potentially establish communications with any website

Atredis recommends having the password change and auto-lock features assessed once they have been implemented.

Browser Extension Overview

Talisman Wallet is a Manifest V2[2] web browser extension built for use in the Google Chrome or Mozilla Firefox web browsers. The primary use case for Talisman is to allow users of Substrate[3]-based blockchains to manage their accounts in one place, with one master key.

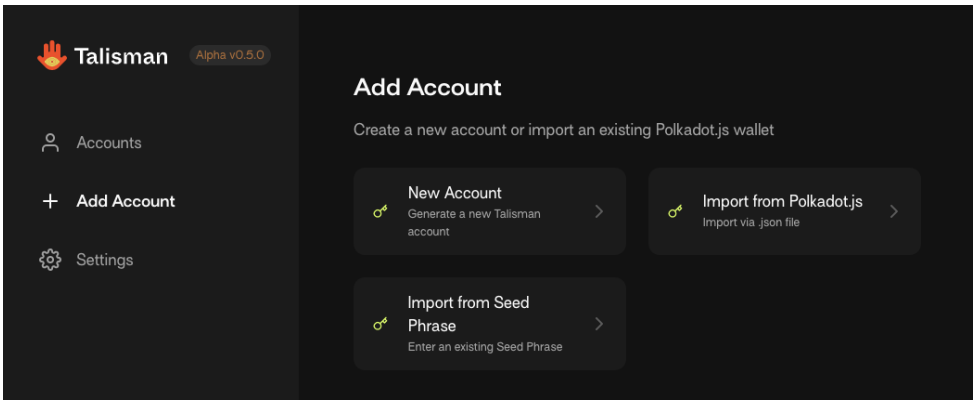
In the version that was tested, the Talisman dashboard listed the master account and any other registered accounts, allowed the user to rename, export, and remove accounts, and provided menu items to add an account and access the extension settings.



Talisman Wallet Dashboard

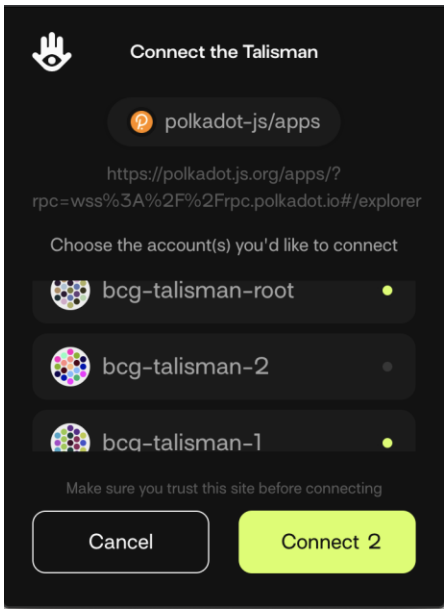
[2] https://developer.chrome.com/docs/extensions/mv2/
[3] https://substrate.io/

From the `Add Account` screen, the user could create a new account managed internally by the wallet, create an account from a mnemonic seed phrase, or import an account from a file in the Polkadot JSON format[4].



Talisman Wallet Add Account Screen

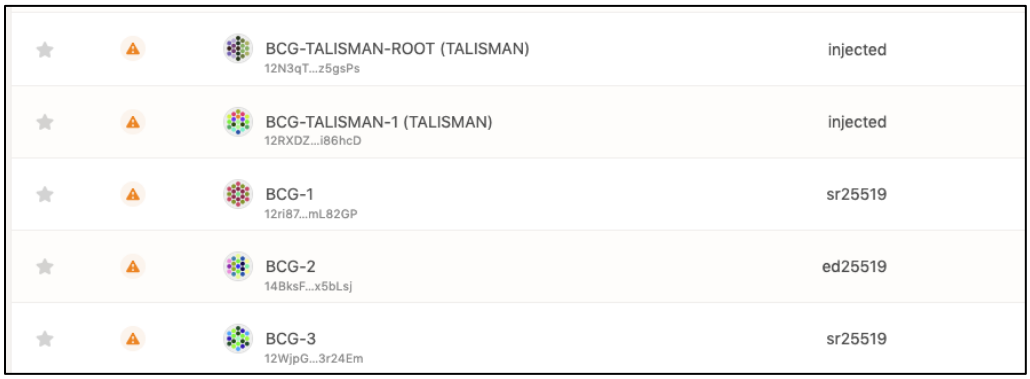
When a page that the extension supported was loaded, the extension would pop-up a dialog asking which accounts the user would like to allow the site to access, as shown below. In this example the `bcg-talisman-root` and `bcg-talisman-1` accounts have been selected to be connected with the Polkadot.js site.



Talisman Account Connection Dialog

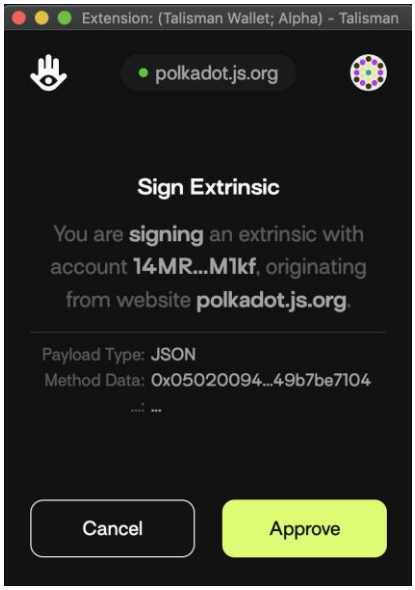
[4] https://wiki.polkadot.network/docs/learn-account-restore

Once the site had been connected to the accounts, the accounts were injected into the site content by the extension. The example below shows how the connected `bcg-talisman-root` and `bcg-talisman-1` accounts were injected into the Polkadot.js account list.



Accounts Injected into the Polkadot.js Account List

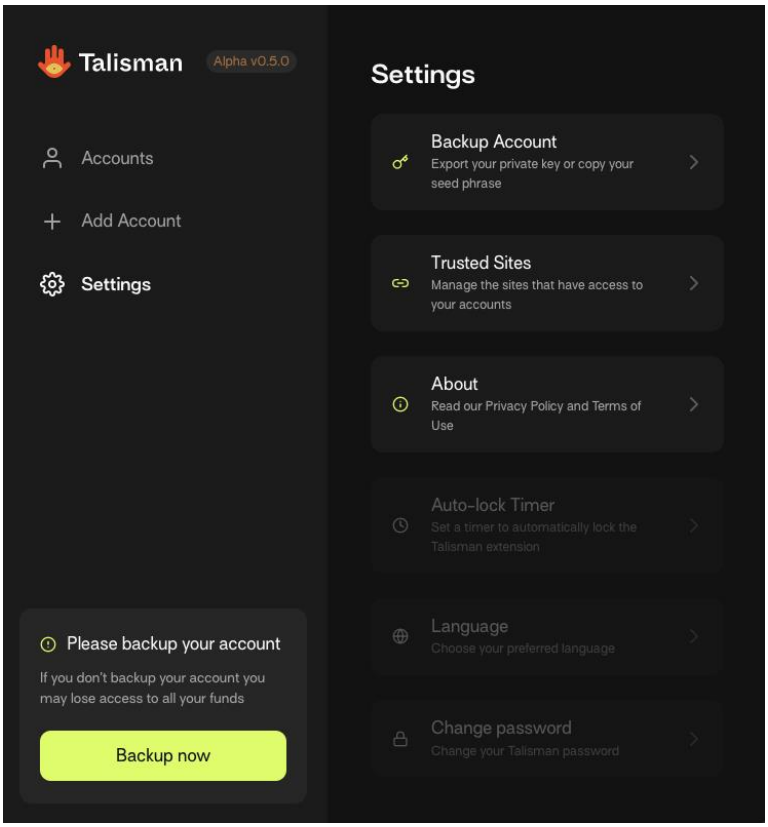
When one of the injected accounts was selected as the source account for an Extrinsics call[5], Talisman would prompt the user to approve the transaction before it would be signed and sent.



Extension Signing Approval Prompt

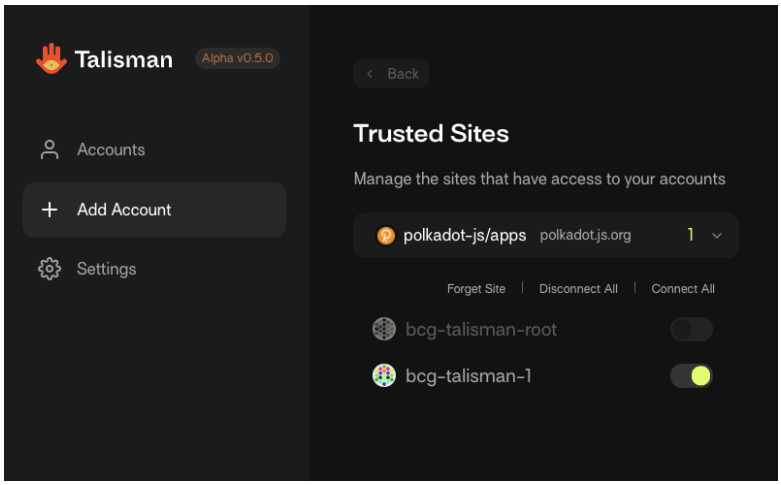
[5] https://polkadot.js.org/docs/substrate/extrinsics/

The Talisman settings screen allowed the user to back up their master key seed phrase and manage trusted sites. The version that was tested also had placeholders for future features to automatically lock the extension and allow the user to change their master password. This functionality had not yet been implemented and as such was not tested during this engagement. Atredis recommends having these features assessed once they have been implemented.



Talisman Settings Screen

The Talisman `Trusted Sites` screen allowed the user to specify which sites can access which accounts.



Talisman Trusted Sites Settings

Stored Data

At the time of testing, the Talisman Wallet stored all of the user's data in the user's local browser extension storage. The extension storage contained the following data:

- A master key "nursery" that was encrypted using the MetaMask Browser-Passworder library[6] which uses AES-GCM[7] and the user's PDBKDF2[8]-derived password as the key
- The user's registered accounts in Polkadot's NaCL[9]-encrypted KeyringJson[10] format, encrypted with the "nursery" key
- Authorized URLs configuration that specifies which sites can access which accounts
- Chain metadata
- The Talisman onboarding state
- A transactions object that did not seem to be used

Extension storage is directly accessible from both the injected content script and the extension background page, but it is not accessible from pages that the extension is injected into. Atredis tested these direct interactions in order to determine the impact of an attacker gaining access to the locked extension storage.

---

[6] https://github.com/MetaMask/browser-passworder
[7] https://www.cryptosys.net/pki/manpki/pki_aesgcmauthencryption.html
[8] https://en.wikipedia.org/wiki/PBKDF2
[9] https://nacl.cr.yp.to/
[10] https://github.com/polkadot-js/ui/tree/master/packages/ui-keyring

After reviewing, analyzing, and discussing the storage structure with the development team, Atredis Partners determined that the storage structure is sound. While there is opportunity to strengthen the storage structure to prevent a local attacker from removing registered accounts and tampering with site authorizations, by storing them inside of the encrypted nursery, it would not prevent the attacker from purging the wallet data entirely.

"Lock" and "Unlock"

Authentication in the Talisman Wallet was handled by prompting the user for their master password and passing it to the extension background page through the `pri(app.onboard)` or `pri(app.authenticate)` extension API operations. The password was only stored in background page memory. Ultimately, if the password variable was set, the user was considered authenticated, and the extension was "unlocked". When the user "locked" the extension, the `pri(app.lock)` API operation was called, which cleared the password variable from memory. This password was used to decrypt the master key, so when the password was not in memory, account private keys could not be accessed.

As part of this assessment, Atredis Partners reviewed the source code, performed dynamic testing, analyzed the workflow within context of the browser controls, and discussed the implementation with the development team. In the end, Atredis determined the authentication mechanism to be an effective means of protecting the user's private keys.

Atredis Partners notes that the Talisman team has plans to implement automatic locking behavior and a password change feature. These features will be essential in providing a full-featured wallet.

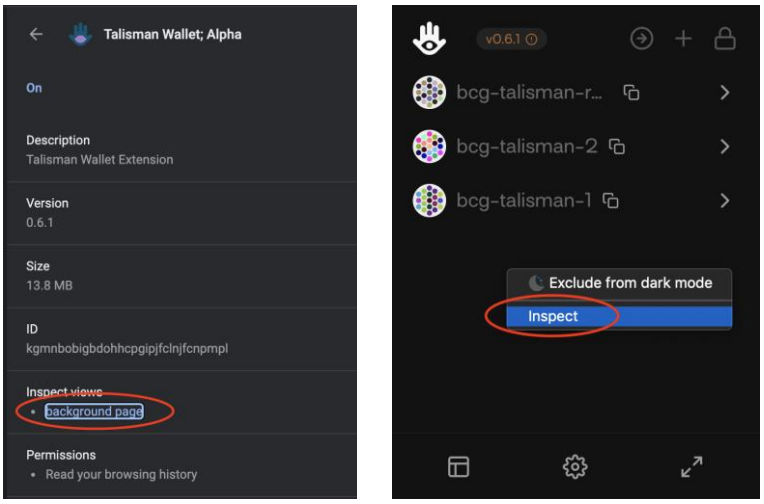
API Operations

The Talisman Wallet background page API is based on the Polkadot.js extension scaffold[11]. Although the structure and the available API operations differ, Talisman uses a similar messaging protocol. On top of the normal ManifestV2 extensions messaging implementation, the Talisman wallet API contains a message routing and type system for handling requests. The API provided 45 callable operations that were related to onboarding, locking and unlocking the extension, opening dashboards and pop-ups, managing accounts, managing site authorization, extrinsic signing, and subscribing to various metadata.

Atredis reviewed the source code and performed dynamic testing of the messaging bootstrap process, message routing, and each API operation to identify potential weaknesses that could be exploited by an attacker who had gained access to the extension API of a locked or unlocked wallet.

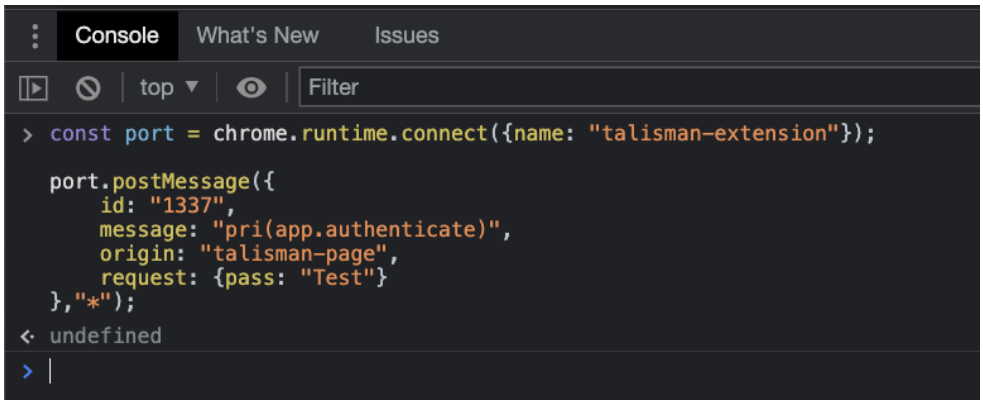
[11] https://github.com/polkadot-js/extension

Dynamic API testing was performed through the web browser consoles. First, the background page console was opened to monitor messages that were passed to and from the background page. Then the pop-up window console was opened to interact with the extension API that is only accessible from within the context of the extension.



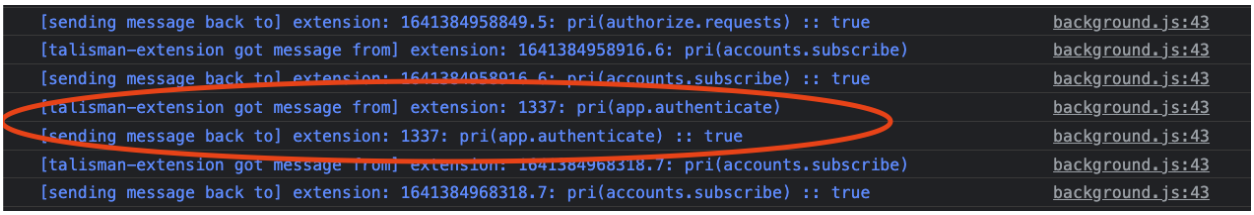
Opening the Background Page and Popup Consoles

From the extension pop-up console, Atredis Partners manually crafted requests to the background page API. The example below shows a request being sent to unlock the extension.



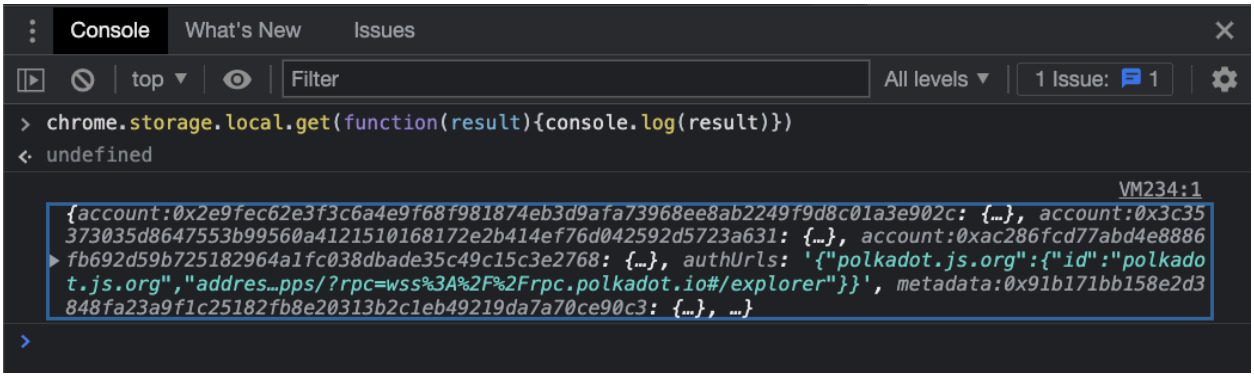
Calling the `pri(app.authenticate)` API Operation

From the background page console, Atredis monitored messages that were sent to and from the API.



Monitoring API Messages: `pri(app.authenticate)`

While Atredis Partners did identify inconsistencies in the API behavior, like the `pri(authorized)` operation returning the list of registered account addresses when the extension was locked, and the ability to call the `pri(account.forget)` operation to drop a linked account when the wallet was locked, it is important to note that an attacker who has access to interact with the extension API also has access to interact the extension storage directly, as shown below. Because of this, issues in the background page API behavior that could be also exploited by directly interacting with the extension storage were not included as findings.



Accessing Extension Storage from Extension Pop-up Context

Extrinsic Signing

One of the key features of the Talisman Wallet extension is to allow extrinsic signing without the user having to type in the unique password for each account they want to perform extrinsic calls with.

When the Talisman extension is activated for a site and a signing request is triggered, the extension pops up the approval screen shown in the overview above. If the user approves the request, the extension backend function `signingApprove` is called. This function is essentially a fork of the `signingApprovePassword` function from the Polkadot.js extension scaffold, with the password prompt removed. In place of the password prompt, the Talisman Wallet extension validates that it can decrypt the selected signing keypair, which fails with the error, "`Password needed to unlock the account`" if the extension is locked. If the extension is unlocked, the signing process proceeds as usual.

Content Scripts

The content scripts are responsible for the main functionality of the Talisman extension. These responsibilities include but are not limited to injecting JavaScript into the visited site and setting up communication between the browser page and the extension.

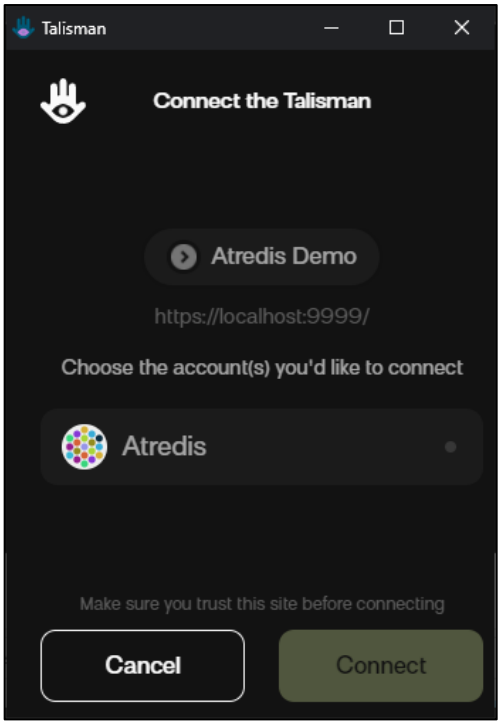
For example, while browsing web pages on the internet, the Talisman extension injects a script tag telling the browser to `GET` a "`page.js`" file hosted by the extension. After the script is loaded, the generated tag is removed. Once the script is loaded and executed by the user's browser, the extension checks if the current URL is a known phishing site. The extension does this by validating the current URL with a list of known bad URLs supplied by `polkadot.js.org`.

If the site is determined to be a phishing site, the user's tab is redirected, and if not, the extension continues to perform its actions as intended. Finally, after the Talisman extension has determined the current URL is not a phishing site, the functionality allowing decentralized applications to communicate with Talisman is enabled.

With the extension running, Talisman utilizes the browser runtime messaging service to perform communication from the browser to the extension. These messages start with `pri` and then contain the action in parentheses, for example, `pri(app.onboard)`. During testing, Atredis Partners identified that the extension was not configured with cross-extension messaging, externally connectable, or native messaging. To confirm, Atredis attempted to send multiple messages from different sources and could not communicate with the extension.

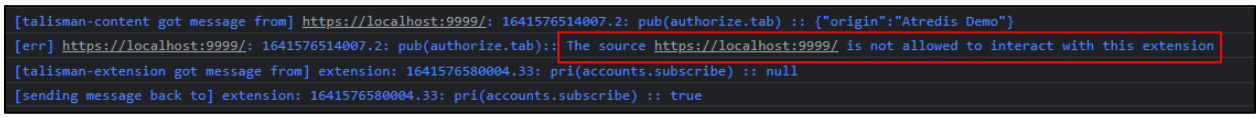
Finally, Atredis Partners tested and observed how the extension interacts with web services visiting a site using the Polkadot.js Web3 library[12]. On the first visit, communication between the library and extension is not permitted until the end-user connects at least one of their accounts. The screenshot below shows the connection pop-up.

[12] https://github.com/polkadot-js/extension/tree/master/packages/extension-dapp



Extension Request

During testing, Atredis was unable to bypass this check and add the site to the "Trusted Sites" or interact further with the Talisman extension if the user denies access.



Extension Access Denied

While a user had to explicitly allow for an application to interact with the Talisman Extension, Atredis Partners determined that it was possible to list previously allowed accounts from a locked state. No additional information or unapproved accounts are disclosed, and it was not possible to make transactions against the approved account while the extension was locked.

Manifest Controls

Atredis reviewed the manifest controls to identify exploitable configuration weaknesses. The key takeaways from the configuration review were:

- A wildcard `content_scripts` `matches` rule is used that allows the extension to be loaded into any HTTPS website. This grants the extension more access than is necessary but it not exploitable.
- A wildcard host permission entry is used that allows the extension to communicate with any HTTPS website. Again, this control grants the extension more permissions than necessary, but is not an exploitable weakness.
- The `storage` permission is used to store accounts in local extension storage. This is expected for extensions that store data locally.
- The `tabs` permission is used for creating a new tab for onboarding and the extension dashboard, and phishing site redirection. This is expected for extensions that provide access to multiple screens.

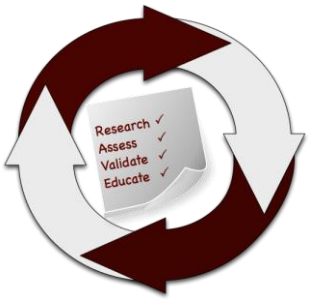
Findings Summary

In performing testing for this assessment, Atredis Partners did not identify any findings.

Atredis defines vulnerability severity ranking as follows:

- **Critical:** These vulnerabilities expose systems and applications to immediate threat of compromise by a dedicated or opportunistic attacker.
- **High:** These vulnerabilities entail greater effort for attackers to exploit and may result in successful network compromise within a relatively short time.
- **Medium:** These vulnerabilities may not lead to network compromise but could be leveraged by attackers to attack other systems or applications components or be chained together with multiple medium findings to constitute a successful compromise.
- **Low:** These vulnerabilities are largely concerned with improper disclosure of information and should be resolved. They may provide attackers with important information that could lead to additional attack vectors or lower the level of effort necessary to exploit a system.

Appendix I: Assessment Methodology



Atredis Partners draws on our extensive experience in penetration testing, reverse engineering, hardware/software exploitation, and embedded systems design to tailor each assessment to the specific targets, attacker profile, and threat scenarios relevant to our client's business drivers and agreed upon rules of engagement.

Where applicable, we also draw on and reference specific industry best practices, regulations, and principles of sound systems and software design to help our clients improve their products while simultaneously making them more stable and secure.

Our team takes guidance from industry-wide standards and practices such as the National Institute of Standards and Technology's (NIST) Special Publications, the Open Web Application Security Project (OWASP), and the Center for Internet Security (CIS).

Throughout the engagement, we communicate findings as they are identified and validated, and schedule ongoing engagement meetings and touchpoints, keeping our process open and transparent and working closely with our clients to focus testing efforts where they provide the most value.

In most engagements, our primary focus is on creating purpose-built test suites and toolchains to evaluate the target, but we do utilize off-the-shelf tools where applicable as well, both for general patch audit and best practice validation as well as to ensure a comprehensive and consistent baseline is obtained.

Research and Profiling Phase

Our research-driven approach to testing begins with a detailed examination of the target, where we model the behavior of the application, network, and software components in their default state. We map out hosts and network services, patch levels, and application versions. We frequently use a number of private and public data sources to collect Open Source Intelligence about the target, and collaborate with client personnel to further inform our testing objectives.

For network and web application assessments, we perform network and host discovery as well as map out all available application interfaces and inputs. For hardware assessments, we study the design and implementation, down to a circuit-debugging level. In reviewing source code or compiled application code, we map out application flow and call trees and develop a solid working understand of how the application behaves, thus helping focus our validation and testing efforts on areas where vulnerabilities might have the highest impact to the application's security or integrity.

Analysis and Instrumentation Phase

Once we have developed a thorough understanding of the target, we use a number of specialized and custom-developed tools to perform vulnerability discovery as well as binary, protocol, and runtime analysis, frequently creating engagement-specific software tools which we share with our clients at the close of any engagement.

We identify and implement means to monitor and instrument the behavior of the target, utilizing debugging, decompilation and runtime analysis, as well as making use of memory and filesystem

forensics analysis to create a comprehensive attack modeling testbed. Where they exist, we also use common off-the-shelf, open-source and any extant vendor-proprietary tools to aid in testing and evaluation.

Validation and Attack Phase

Using our understanding of the target, our team creates a series of highly-specific attack and fault injection test cases and scenarios. Our selection of test cases and testing viewpoints are based on our understanding of which approaches are most relevant to the target and will gain results in the most efficient manner, and built in collaboration with our client during the engagement.

Once our test cases are validated and specific attacks are confirmed, we create proof-of-concept artifacts and pursue confirmed attacks to identify extent of potential damage, risk to the environment, and reliability of each attack scenario. We also gather all the necessary data to confirm vulnerabilities identified and work to identify and document specific root causes and all relevant instances in software, hardware, or firmware where a given issue exists.

Education and Evidentiary Phase

At the conclusion of active testing, our team gathers all raw data, relevant custom toolchains, and applicable testing artifacts, parses and normalizes these results, and presents an initial findings brief to our clients, so that remediation can begin while a more formal document is created. Additionally, our team shares confirmed high-risk findings throughout the engagement so that our clients may begin to address any critical issues as soon as they are identified.

After the outbrief and initial findings review, we develop a detailed research deliverable report that provides not only our findings and recommendations but also an open and transparent narrative about our testing process, observations and specific challenges in developing attacks against our targets, from the real world perspective of a skilled, motivated attacker.

Automation and Off-The-Shelf Tools

Where applicable or useful, our team does utilize licensed and open-source software to aid us throughout the evaluation process. These tools and their output are considered secondary to manual human analysis, but nonetheless provide a valuable secondary source of data, after careful validation and reduction of false positives.

For runtime analysis and debugging, we rely extensively on Hopper, IDA Pro and Hex-Rays, as well as platform-specific runtime debuggers, and develop fuzzing, memory analysis, and other testing tools primarily in Ruby and Python.

In source auditing, we typically work in Visual Studio, Xcode and Eclipse IDE, as well as other markup tools. For automated source code analysis we will typically use the most appropriate toolchain for the target, unless client preference dictates another tool.

Network discovery and exploitation make use of Nessus, Metasploit, and other open-source scanning tools, again deferring to client preference where applicable. Web application runtime analysis relies extensively on the Burp Suite, Fuzzer and Scanner, as well as purpose-built automation tools built in Go, Ruby and Python.

Engagement Deliverables

Atredis Partners deliverables include a detailed overview of testing steps and testing dates, as well as our understanding of the specific risk profile developed from performing the objectives of the given engagement.

In the engagement summary we focus on "big picture" recommendations and a high-level overview of shared attributes of vulnerabilities identified and organizational-level recommendations that might address these findings.

In the findings section of the document, we provide detailed information about vulnerabilities identified, provide relevant steps and proof-of-concept code to replicate these findings, and our recommended approach to remediate the issues, developing these recommendations collaboratively with our clients before finalization of the document.

Our team typically makes use of both DREAD and NIST CVE for risk scoring and naming, but as part of our charter as a client-driven and collaborative consultancy, we can vary our scoring model to a given client's preferred risk model, and in many cases will create our findings using the client's internal findings templates, if requested.

Sample deliverables can be provided upon request, but due to the highly specific and confidential nature of Atredis Partners' work, these deliverables will be heavily sanitized, and give only a very general sense of the document structure.

Appendix II: Engagement Team Biographies

Shawn Moyer, Founding Partner and CEO

Shawn Moyer scopes, plans, and coordinates security research and consulting projects for the Atredis Partners team, including reverse engineering, binary analysis, advanced penetration testing, and private vulnerability research. As CEO, Shawn works with the Atredis leadership team to build and grow the Atredis culture, making Atredis Partners a home for some of the best minds in information security, and ensuring Atredis continues to deliver research and consulting services that exceed our client's expectations.

Experience

Shawn brings over 25 years of experience in information security, with an extensive background in penetration testing, advanced security research including extensive work in mobile and Smart Grid security, as well as advanced threat modeling and embedded reverse engineering.

Shawn has served as a team lead and consultant in enterprise security for numerous large initiatives in the financial sector and the federal government, including IBM Internet Security Systems' X-Force, MasterCard, a large Federal agency, and Wells Fargo Securities, all focusing on emerging network and application attacks and defenses.

In 2010, Shawn created Accuvant Labs' Applied Research practice, delivering advanced research-driven consulting to numerous clients on mobile platforms, critical infrastructure, medical devices and countless other targets, growing the practice 1800% in its first year.

Prior to Accuvant, Shawn helped develop FishNet Security's penetration testing team as a principal security consultant, growing red team offerings and advanced penetration testing services, while being twice selected as a consulting MVP.

Key Accomplishments

Shawn has written on emerging threats and other topics for Information Security Magazine and ZDNet, and his research has been featured in the Washington Post, BusinessWeek, NPR and the New York Times. Shawn is a twelve-time speaker at the Black Hat Briefings and has been an invited speaker at other notable security conferences around the world.

Shawn is likely best known for delivering the first public research on social network security, pointing out much of the threat landscape still exists on social network platforms today. Shawn also co-authored an analysis of the state of the art in web browser exploit mitigation, creating the first in-depth comparison of browser security models along with Dr. Charlie Miller, Chris Valasek, Ryan Smith, Joshua Drake, and Paul Mehta.

Shawn studied Computer and Network Information Systems at Missouri University and the University of Louisiana at Lafayette, holds numerous information security certifications, and has been a frequent presenter at national and international security industry conferences.

Bryan C. Geraghty, Principal Research Consultant

Bryan leads and executes highly technical application and network security assessments, as well as adversarial simulation assessments. He specializes in cryptography and reverse engineering.

Experience

Bryan has over 20 years of experience building and exploiting networks, software, and hardware systems. His deep background in systems administration, software development, and cryptography has been demonstrably beneficial for security assessments of custom or unique applications in industries such as healthcare, manufacturing, marketing, banking, utilities, and entertainment.

Key Accomplishments

Bryan is a creator and maintainer of several open-source security tools. He is also a nationally recognized speaker; often presenting research on topics such as software, hardware, and communications protocol attacks, and participating in offense-oriented panel discussions. Bryan is also an organizing-board member of multiple Kansas City security events, and a staff volunteer & organizer of official events at DEF CON.

James Cook, Senior Research Consultant

James executes highly technical software security, network, and web application assessments and advanced red team assessments.

Experience

James has over 6 years of experience in the information security industry. James has developed and released numerous open-source security tools that are used by many information security professionals. He holds the Offensive Security Certified Professional (OSCP) certification and has performed a wide variety of security assessments including network penetration testing, application security assessments, full-scope red team engagements, adversarial simulation, and physical penetration testing

Prior to joining Atredis Partners, James performed network penetration tests as a Senior Security Consultant on Optiv's Attack and Penetration team.

Key Accomplishments

James has presented at Black Hat Arsenal and DerbyCon. James has also contributed to the Metasploit Framework and Veil Evasion Toolkit, as well as several other open source security tools.

Molly Vukusich, Client Operations Associate

Molly Vukusich supports nearly every phase of the project lifecycle at Atredis Partners, from pre-sales, to project planning and management, to project delivery, readout and follow-up. She aims to increase efficiency of project execution and client communication for the benefit of both the consultants and clients.

Experience

Molly has over 11 years of experience in marketing and project management roles in various industries such as Healthcare, Finance, Sports & Recreation, and Non-Profit. Her experience includes copywriting and editing (both technical and promotional), creative strategy development, data analysis, event planning, graphic design, and website management.

Key Accomplishments

Molly earned a bachelor's degree in Mass Communications with an emphasis in Advertising and Public Relations from Oklahoma City University.

Tom Steele, Research Consulting Director

Tom Steele leads and executes application security assessments and adversarial engagements, ranging from source code review to advanced red team assessments.

Experience

Tom has over eight years of professional experience in information security. During that time, his focus has been on executing and innovating both network and application level assessments; with a focus on developing new techniques, tools, and processes that improve collaborative testing, coverage, deterrent bypass, and data exfiltration.

In addition to performing assessments, Tom is also a seasoned software developer, and has an expert knowledge of multiple languages and platforms including Go and Node.js. Tom understands how applications fit together and has used his development experience to develop and maintain many widely used open-source and proprietary tools including Lair, a real-time testing collaboration application, and BurpBuddy, an API for BurpSuite Pro.

Prior to joining Atredis, Tom was a practice manager on Optiv's Attack and Penetration team, where he led a team of consultants, developed and enhanced methodologies, toolsets, and processes, and conducted hundreds of security assessments.

Key Accomplishments

Tom is a contributor to the Node Security Project, where he has assisted with the identification and remediation of many vulnerabilities; both in Node core and in widely deployed libraries. He has consulted leaders working at Fortune 500 companies on how to increase the security of their application frameworks. He has presented and lead training at several conferences including Black Hat, DEF CON, BSides, and DerbyCon and is the Co-Author of No Starch Press' "Black Hat Go".

Lacey Kasten, Client Operations, Technical Writer and Editor

Lacey Kasten helps facilitate client operations and deliverable creation/development at Atredis Partners. She supports pre-sales project scoping and back-end operations efforts, shepherding of the technical writing style and voice at Atredis, and the final quality assurance review of penetration test deliverables prior to engagement completion. Lacey seeks to provide readable, understandable communication to Atredis Partners' clientele. She stays embedded in the Information Security community and is passionate about accessible and equitable knowledge transfer in all mediums across a wide span of Cyber Security, Threat Intelligence, National Security, and Open Source topics.

Experience

Lacey has worked in communications roles from within the Fine Art and Design industry, Museum and Non-Profit Philanthropy space, Biomedical Computer Science, Higher Education Public Relations, and Event and Tradeshow industry throughout her career. Her work spans writing (technical, copy editing, social media marketing, and blogging), editing and mentorship of writers in the Information Security space, content creation (web development, event planning, graphic design, and photography), and film and movie production.

Key Accomplishments

Lacey achieved a bachelor's degree in Communication Design from the Pacific Northwest College of Art in Portland, Oregon. Lacey has contributed to Open Source Intelligence (OSINT) tool projects and participated in testing, documenting, and in project management for other Open Source development projects. Currently, Lacey supports the FLOSS (Free/Libre/Open Source Software) community by serving on the core organizing staff of SeaGL (Seattle GNU/Linux) conference. Previously, she was a member on the Board of Directors for the largest Information Security conference in the United States Pacific Northwest, Security BSides PDX, and served the charitable 501(c)(3) as coordinator of Sponsorship and Endowment.

Appendix III: About Atredis Partners

Atredis Partners was created in 2013 by a team of security industry veterans who wanted to prioritize offering quality and client needs over the pressure to grow rapidly at the expense of delivery and execution. We wanted to build something better, for the long haul.

In six years, Atredis Partners has doubled in size annually, and has been named three times to the Saint Louis Business Journal's "Fifty Fastest Growing Companies" and "Ten Fastest Growing Tech Companies". Consecutively for the past three years, Atredis Partners has been listed on the Inc. 5,000 list of fastest growing private companies in the United States.

The Atredis team is made up of some of the greatest minds in Information Security research and penetration testing, and we've built our business on a reputation for delivering deeper, more advanced assessments than any other firm in our industry.

Atredis Partners team members have presented research over forty times at the BlackHat Briefings conference in Europe, Japan, and the United States, as well as many other notable security conferences, including RSA, ShmooCon, DerbyCon, BSides, and PacSec/CanSec. Most of our team hold one or more advanced degrees in Computer Science or engineering, as well as many other industry certifications and designations. Atredis team members have authored several books, including *The Android Hacker's Handbook*, *The iOS Hacker's Handbook*, *Wicked Cool Shell Scripts*, *Gray Hat C#*, and *Black Hat Go*.

While our client base is by definition confidential and we often operate under strict nondisclosure agreements, Atredis Partners has delivered notable public security research on improving the security at Google, Microsoft, The Linux Foundation, Motorola, Samsung and HTC products, and were the first security research firm to be named in Qualcomm's Product Security Hall of Fame. We've received four research grants from the Defense Advanced Research Project Agency (DARPA), participated in research for the CNCF (Cloud Native Computing Foundation) to advance the security of Kubernetes, worked with OSTIF (The Open Source Technology Improvement Fund) and The Linux Foundation on the Core Infrastructure Initiative to improve the security and safety of the Linux Kernel, and have identified entirely new classes of vulnerabilities in hardware, software, and the infrastructure of the World Wide Web.

In 2015, we expanded our services portfolio to include a wide range of advanced risk and security program management consulting, expanding our services reach to extend from the technical trenches into the boardroom. The Atredis Risk and Advisory team has extensive experience building mature security programs, performing risk and readiness assessments, and serving as trusted partners to our clients to ensure the right people are making informed decisions about risk and risk management.