SHOW NOTES

@mjkabir Notes

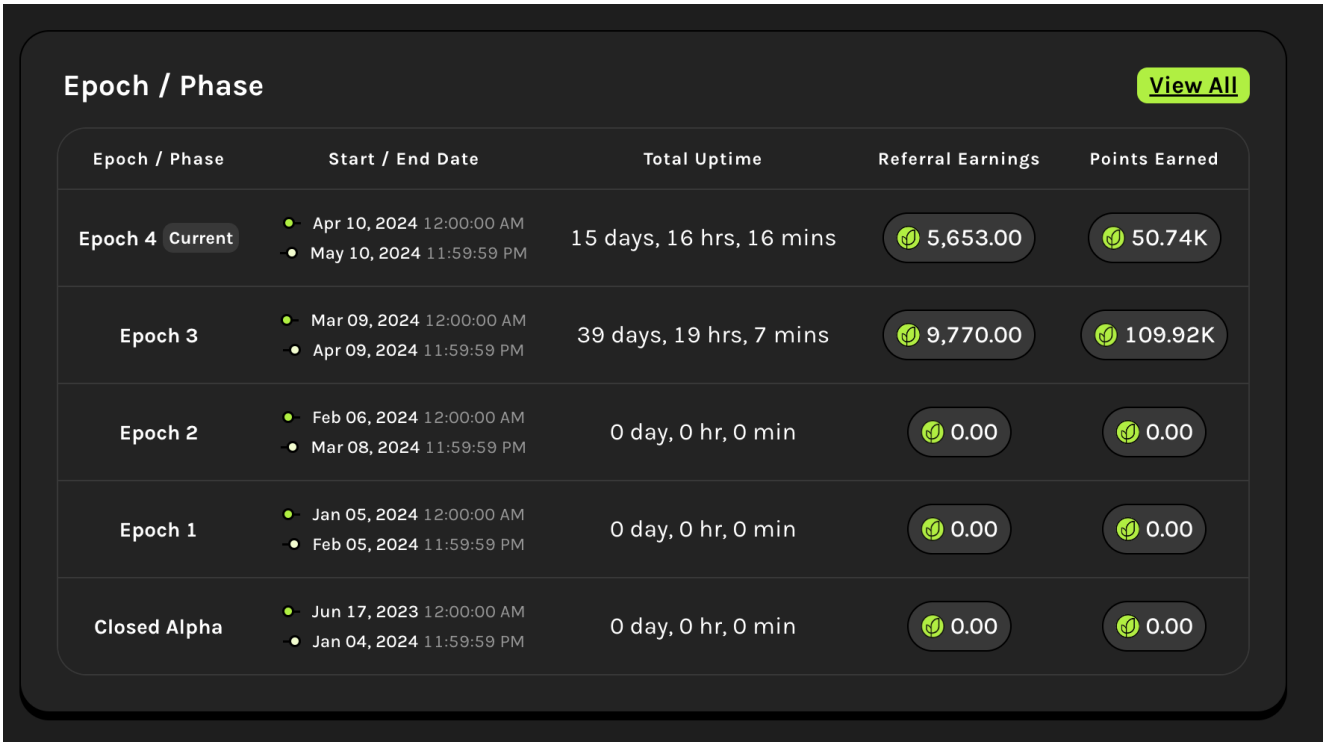
https://shownotes.app/show/fuhwb

Grass Notes

I have been exploring the Grass project by Wynd Labs in Canada. All my notes related to this project can be found here. I like this project, so I wanted to provide as many insights as possible for anyone to explore this. Do your own research.

DISCLAIMER

If you join Grass to become a node by installing the Chrome / Brave plugin, you do so at your own risk. I have provided my views, and I do participate in the Grass network, but I am not responsible for your decisions.



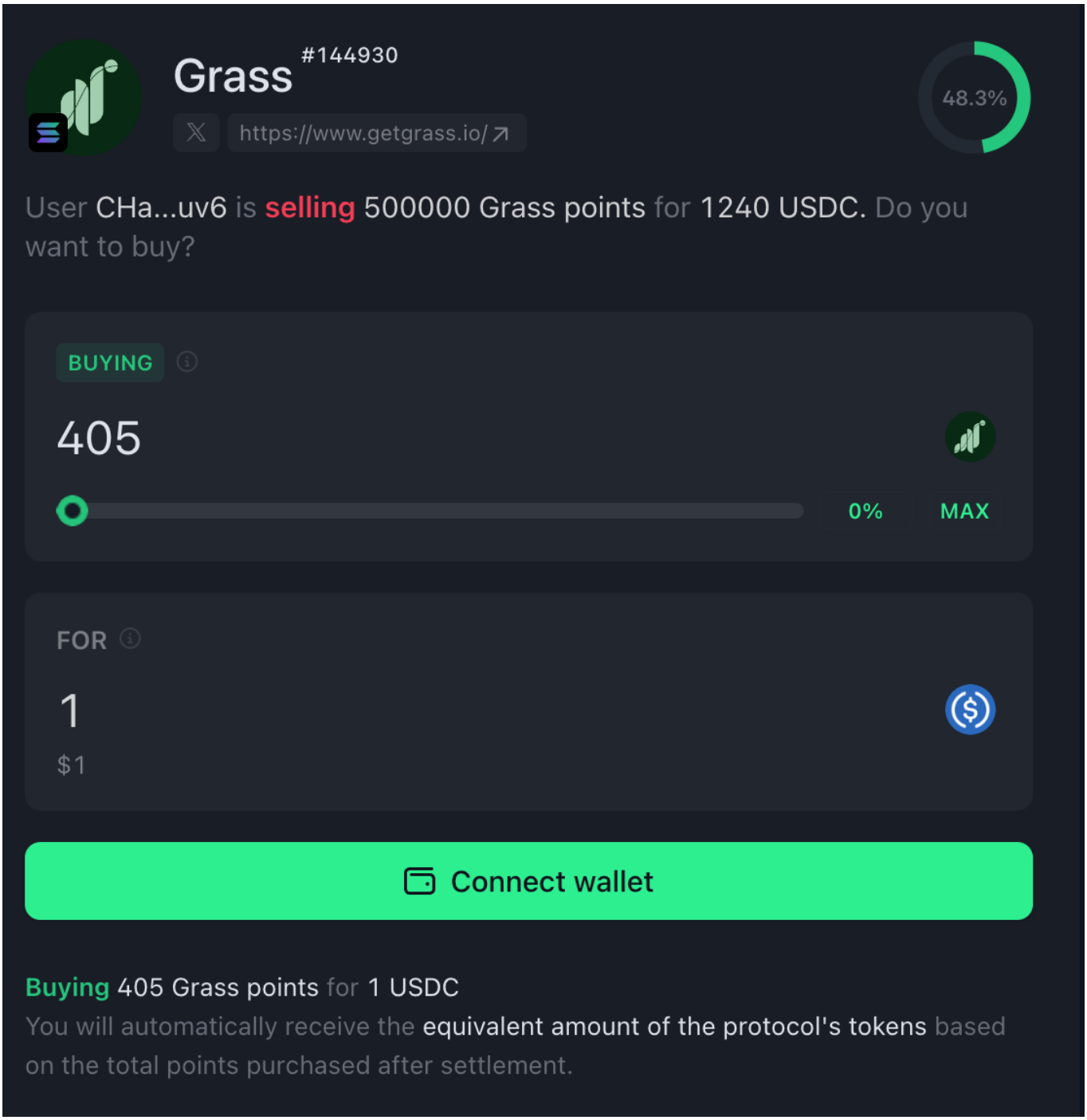
AI REVIEW PASSED.

Trade Grass Points in WhalesMarket

Some folks have asked me about how to monetize the grass rewards. Personally, I have no rush and I will accumulate until the team finally clarify how to redeem the rewards.

But appearntly, some people are trading their grass rewards on this marketplace. I have not used WhaleMarket for anything so I do not have practical experience using it. But from what I see is that people who have got grass reward are somehow able to transfer it to this platform and able to put them up for an asking price (in $USDC stable coin in the Solana network).

For example, here is an offer that is being partically filled. This person is selling 405 grass rewards for 1 $USDC (equiv. of 1 US dollar).



This is very interesting. At the time of writing this note, I have 50K grass rewards accumulated. So this would mean that it would be sellable at (50,000 grass / 405 grass) = 123 $USDC. Now I have accumulated this amount over last 20 days so, you can say that 123 $USDC / 20 days = 6.15 $USD per day. This indicates that in a month, I can accumulate 184.5 $USDC worth of grass. Again all these calculation is based on this example. The actual grass reward conversion to $USDC or $SOL will vary based on how the project works out. But this gives you an idea of how things are shaping up.

578 days 15 hrs ago

Website:

https://app.whales.market/points-markets?project=Grass

Security Review of Grass Chrome/Brave Extension

WebSockets Communication

The extension uses WebSockets to communicate with external servers.
The URLs wss://proxy.wynd.network:4650 and wss://proxy.wynd.network:4444 are hardcoded, which could be risky if the server is not secure or compromised.

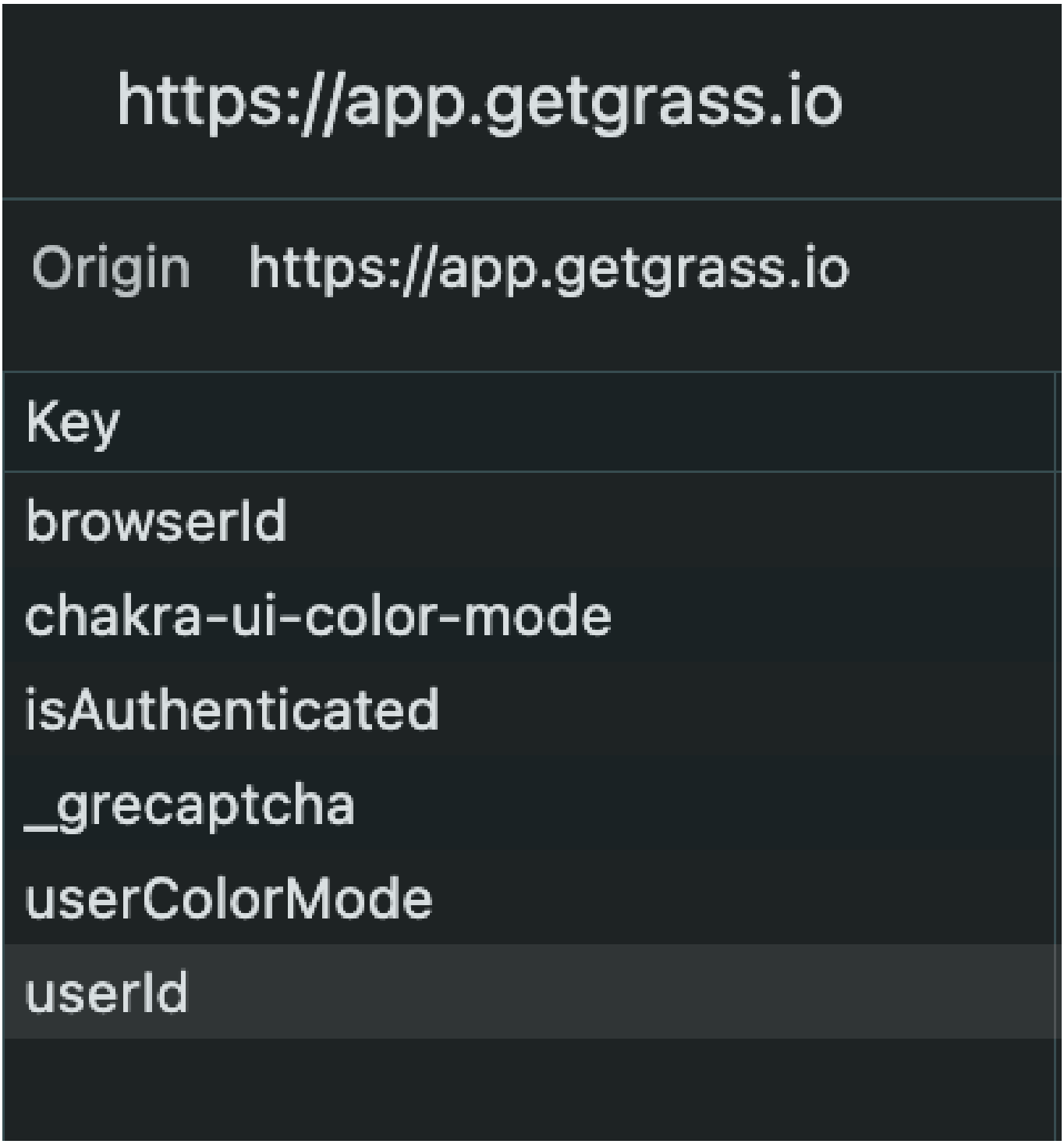
Additionally, if the WebSocket connection is not properly secured, it could be vulnerable to man-in-the-middle attacks.

External Script Interaction

The extension listens for messages from external scripts (onMessageExternal listener). This can be risky if the extension does not validate the origin of the messages properly, leading to potential external control over the extension's behavior.

Local Storage Usage

The extension stores and retrieves sensitive information such as JWT tokens, user IDs, and settings in Chrome's local storage (chrome.storage.local). This data might be accessible to other scripts running in the context of the extension's background page, posing a risk if any of those scripts are compromised.



Header Manipulation
The extension modifies request headers (HEADERS_TO_REPLACE list and onBeforeRequest listener). Improper manipulation of headers like Cookie, Referer, and others could lead to security vulnerabilities, including session fixation, cross-site request forgery (CSRF), and others.

Error Handling and Logging
The extension logs errors and other information using LogsTransporter.sendLogs. Depending on the implementation of this logging mechanism, sensitive information could be inadvertently exposed to an external server.

Use of eval-like Functions: The code uses JSON.parse in several places without proper input validation. Malicious input could lead to injection attacks if not handled properly.

Permissions and Content Security Policy (CSP)

The extension requires broad permissions ( ) and interacts with permissions dynamically. If the extension's CSP is not restrictive enough, or if permissions are not handled securely, it could lead to vulnerabilities.

Lack of Encryption for Sensitive Data

While the extension uses WebSockets Secure (WSS), the handling and storage of sensitive data such as tokens and user IDs do not explicitly mention encryption, which could be a risk if this data is stored or transmitted insecurely.

602 days 1 hr ago