TxPipe
SHOP

December 7th, 2023

DexHunter - DCA

Contents

1 - Summary

We undertook a thorough examination of DexHunter's DCA (dollar cost averaging) validator. DexHunter, a decentralized exchange (DEX) aggregator, emphasizes a high quality user interface (UI) and user experience (UX), orchestrating a suite of features to enhance liquidity provision and order execution.

The audit is conducted without warranties or guarantees of the quality or security of the code. The investigation spanned several potential vulnerabilities, including scenarios where attackers might exploit the validator to lock up or steal funds. It's important to note that this report only covers identified issues, and we do not claim to have detected all potential vulnerabilities.

1.a - Overview

DexHunter endeavors to extend their capabilities by executing DCA orders without requiring the interim holding of Cardano Native Assets by the DexHunter team.

The process of incrementally investing set amounts over regular intervals is streamlined, requiring only the transfer of assets along with a datum to the validator's address. The validator is programmed to handle two critical actions when checking whether a DCA UTXO (Unspent Transaction Output) can be spent, namely: cancel and execute.

- **Cancel**: In a cancel operation, the validator checks the transaction to ensure it is signed by the owner, as specified in the datum.
- **Execute**: In an execution scenario, the validator needs to check that the outputs are sent to the DEX order contracts with the proper datum and after the correct interval. It also checks that the remaining funds are sent back to the DCA contract for further executions if there are remaining executions as specified in the datum. If there are no remaining executions, the validator checks that the owner of the assets is issued a proper full refund. Finally the validator verifies that the transaction is authenticated by the backend's signature, again as dictated by the datum.

This architecture endows users with a robust mechanism to perform DCA investing on Cardano Native Assets, enhancing their trading strategy while maintaining a trustless ethos by obviating the need for an intermediary to hold assets.

1.b - Process

The audit process involved a meticulous review of the DCA validator for the DexHunter platform. Our team focused on scrutinizing areas susceptible to potential security threats, where attackers could exploit the contract's functions to lock up or steal funds from the dApp and its users. Since this validator involves a trusted backend to execute buys on an interval we also took care to identify ways the backend can abuse it's vital role. Our methodical approach encompassed a range of potential attack vectors, including unauthorized minting, funds theft, denial of service, and business requirements violation, among others. The audit, carried out from Nov 28 to Dec 7, 2023, involved regular interactions on Discord where feedback was submitted. The DexHunter team addressed all issues, as detailed in this report.

1.b.a - Files Audited

Below is a list of all files audited in this report, any files **not** listed here were **not** audited. The final state of the files for the purposes of this report is considered to be commit `88005f9e8a2a9dba5318acd8bde186737890e5e4`.

Filename
validators/dca.ak

2 - Findings

ID	Title	Severity	Status
DH-001	Should not use list.last to get `change_output`	Critical	Resolved
DH-002	Datum should include the assets it's buying	Critical	Resolved
DH-003	Outputs don't prevent dust attacks allowing assets to be locked forever	Critical	Resolved
DH-004	Bound check was using `contains` instead of `entirely_after`	Critical	Resolved
DH-005	Remaining executions was not checked by the smart contract	Critical	Resolved
DH-006	Outputs to dexes should check receiver address is present in the datum	Critical	Resolved
DH-007	Remainder output full address not being checked on DCA order	Critical	Resolved
DH-101	On refund make sure output includes full owner receiver address	Major	Resolved
DH-301	`change_output` is a misleading name	Info	Resolved
DH-302	`must_be_a_valid_order` should destructure the datum eagerly	Info	Resolved

3 - DH-001 Should not use list.last to get `change_output`

Category	Commit	Severity	Status
Exploit	efcc05ff492ca5f5f5d1fba524fccd83b4f09d9e	Critical	Resolved

3.a - Description

`list.last` can be exploited with double satisfaction. This would allow multiple validators to validate the same `change_output`.

3.b - Recommendation

Switch the `change_output` to be directly after the tagged unique output.

3.c - Resolution

Resolved in commit `68c4858bae7361a744695923e624d087a5af3487`

4 - DH-002 Datum should include the assets it's buying

Category	Commit	Severity	Status
Exploit	efcc05ff492ca5f5f5d1fba524fccd83b4f09d9e	Critical	Resolved

4.a - Description

The datum in the output that goes to DEX contracts should include the assets it's buying otherwise the backend can change the assets being bought.

4.b - Recommendation

Take the bytes of the datum that goes to the DEX contract and slice off the chunk that includes the owner and the assets being bought. Then hash that and compare it to the hash in the DCA datum.

4.c - Resolution

Resolved in commit `7ba0c315b81bef4026d7f7206483d394458768f8`

5 - DH-003 Outputs don’t prevent dust attacks allowing assets to be locked forever

Category	Commit	Severity	Status
Exploit	efcc05ff492ca5f5f5d1fba524fccd83b4f09d9e	Critical	Resolved

5.a - Description

The outputs need to be checked to prevent unnecessary assets from being included in them otherwise the backend can lock up a user’s assets forever.

5.b - Recommendation

Check for exact values in the outputs.

5.c - Resolution

Resolved in commit `68c4858bae7361a744695923e624d087a5af3487`

6 - DH-004 Bound check was using `contains` instead of `entirely_after`

Category	Commit	Severity	Status
Exploit	efcc05ff492ca5f5f5d1fba524fccd83b4f09d9e	Critical	Resolved

6.a - Description

The interval bounds check allowed for any validity interval to be valid allowing for the backend to DCA any number of times consecutively.

6.b - Recommendation

Switch to using `entirely_after` instead of `contains`.

6.c - Resolution

Resolved in commit `68c4858bae7361a744695923e624d087a5af3487`

7 - DH-005 Remaining executions was not checked by the smart contract

Category	Commit	Severity	Status
Exploit	efcc05ff492ca5f5f5d1fba524fccd83b4f09d9e	Critical	Resolved

7.a - Description

Remaining executions amount was not being checking allowing the backend to execute more DCA orders than the user intended.

7.b - Recommendation

Check that the remaining executions is greater than 0.

7.c - Resolution

Resolved in commit `68c4858bae7361a744695923e624d087a5af3487`

8 - DH-006 Outputs to dexes should check receiver address is present in the datum

Category	Commit	Severity	Status
Exploit	efcc05ff492ca5f5f5d1fba524fccd83b4f09d9e	Critical	Resolved

8.a - Description

The datum at the output that goes to DEX contracts should include the receiver that's buying assets otherwise the backend can change where the purchased assets go.

8.b - Recommendation

Take the bytes of the datum that goes to the DEX contract and slice off the chunk that includes the owner and the assets being bought. Then hash that and compare it to the hash in the DCA datum.

8.c - Resolution

Resolved in commit `7ba0c315b81bef4026d7f7206483d394458768f8`

9 - DH-007 Remainder output full address not being checked on DCA order

Category	Commit	Severity	Status
Exploit	efcc05ff492ca5f5f5d1fba524fccd83b4f09d9e	Critical	Resolved

9.a - Description

Without checking the remainder output address when placing DCA orders the backend could place a DCA order and steal the remaining funds.

9.b - Recommendation

Enforce that the script address is the address of the remainder output.

9.c - Resolution

Resolved in commit `88005f9e8a2a9dba5318acd8bde186737890e5e4`

10 - DH-101 On refund make sure output includes full owner receiver address

Category	Commit	Severity	Status
Exploit	efcc05ff492ca5f5f5d1fba524fccd83b4f09d9e	Major	Resolved

10.a - Description

Without checking for the stake key credential the backend can take control of where another person's assets are staked when issuing a full refund and therefore can later claim the staking rewards.

10.b - Recommendation

Enforce that the full owner receiver address is the address of the remainder output.

10.c - Resolution

Resolved in commit `68c4858bae7361a744695923e624d087a5af3487`

11 - DH-301 `change_output` is a misleading name

Category	Commit	Severity	Status
Readability	efcc05ff492ca5f5f5d1fba524fccd83b4f09d9e	Info	Resolved

11.a - Description

Bad name for the variable, a more accurate name is `remainder_output`. Usually change output refers to the transaction change output for the creator of the transaction. This variable actually represents the remaining output that contains assets left over from a DCA execution.

11.b - Recommendation

Change `change_output` to `remainder_output`.

11.c - Resolution

Resolved in commit `68c4858bae7361a744695923e624d087a5af3487`

12 - DH-302 `must_be_a_valid_order` should destructure the datum eagerly

Category	Commit	Severity	Status
Optimize	efcc05ff492ca5f5f5d1fba524fccd83b4f09d9e	Info	Resolved

12.a - Description

Field access isn't free and so situation where field access is repeated should be avoided if possible.

12.b - Recommendation

Destructure the datum eagerly to avoid repeated field access.

12.c - Resolution

Resolved in commit `f8c80d1ecd3345f1f2eec89cc1743d25aa021e76`

13 - Appendix

13.a - Disclaimer

This report is governed by the terms in the agreement between TxPipe (**TXPIPE**) and DexHunter (**CLIENT**). This report cannot be shared, referred to, altered, or relied upon by any third party without TXPIP's written consent. This report does not endorse or disapprove any specific project, team, code, technology, asset or similar. It provides no warranty or guarantee about the quality or nature of the technology analyzed.

TXPIPE DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, related to this report, its content, and the related services and products. This report is provided as-is. TxPipe does not take responsibility for any product or service advertised or offered by Client or any third party. **TXPIPE IS NOT RESPONSIBLE FOR MONITORING ANY TRANSACTION BETWEEN YOU AND CLIENT AND/OR ANY THIRD-PARTY PROVIDERS OF PRODUCTS OR SERVICES.**

This report should not be used for making investment or involvement decisions with any project, services or assets. This report provides general information and is not a form of financial, investment, tax, legal, regulatory, or other advice.

TxPipe created this report as an informational review of the due diligence performed on the Client's smart contract. This report provides no guarantee on the security or operation of the smart contract on deployment or post-deployment. **TXPIPE HAS NO DUTY TO MONITOR CLIENT'S OPERATION OF THE PROJECT AND UPDATE THE REPORT ACCORDINGLY.**

The information in this report may not cover all vulnerabilities. This report represents an extensive assessment process intended to help increase the quality of the Client's code. However, blockchain technology and cryptographic assets present a high level of ongoing risk, including unknown risks and flaws.

TxPipe recommends multiple independent audits, a public bug bounty program, and continuous security auditing and monitoring. Errors in the manual review process are possible, and TxPipe advises seeking multiple independent opinions on critical claims. **TXPIPE BELIEVES EACH COMPANY AND INDIVIDUAL IS RESPONSIBLE FOR THEIR OWN DUE DILIGENCE AND CONTINUOUS SECURITY.**

13.b - Issue Guide

13.b.a - Severity

Severity	Description
Critical	Critical issues highlight exploits, bugs, loss of funds, or other vulnerabilities that prevent the dApp from working as intended. These issues have no workaround.
Major	Major issues highlight exploits, bugs, or other vulnerabilities that cause unexpected transaction failures or may be used to trick general users of the dApp. dApps with Major issues may still be functional.
Minor	Minor issues highlight edge cases where a user can purposefully use the dApp in a non-incentivized way and often lead to a disadvantage for the user.
Info	Info are not issues. These are just pieces of information that are beneficial to the dApp creator. These are not necessarily acted on or have a resolution, they are logged for the completeness of the audit.

13.b.b - Status

Status	Description
Resolved	Issues that have been **fixed** by the **project** team.
Acknowledged	Issues that have been **acknowledged** or **partially fixed** by the **project** team. Projects can decide to not **fix** issues for whatever reason.
Identified	Issues that have been **identified** by the **audit** team. These are waiting for a response from the **project** team.

13.c - Revisions

This report was created using a git based workflow. All changes are tracked in a github repo and the report is produced using typst. The report source is available here. All versions with downloadable PDFs can be found on the releases page.

13.d - About Us

TxPipe is a blockchain technology company responsible for many projects that are now a critical part of the Cardano ecosystem. Our team built Oura, Scrolls, Pallas, Demeter, and we're the original home of Aiken. We're passionate about making tools that make it easier to build on Cardano. We believe that blockchain adoption can be accelerated by improving developer experience. We develop blockchain tools, leveraging the open-source community and its methodologies.

13.d.a - Links

- Website
- Email
- Twitter