Haven Protocol
Private Decentralized Finance

Technical Overview of June 2021 Exploits

July 7, 2021

Authors: David Bandtock, Neil Coggins, Mattyk

Version 1.3

Document Details

This is a living document that will be updated as new details come to light pending an ongoing investigation. All timestamps are UTC.

Versions

Version	*Date*	*Author/Editor*	*Change*
1.0	July 5, 2021	Mattyk	First draft
1.1	July 7, 2021	David	Sanity check
1.2	July 7, 2021	Neil	Technical sign off
1.3	July 8, 2021	AHawk	Minor edits

Introduction

Starting from June 22, 2021, hackers attacked Haven Protocol, exploiting several related vulnerabilities. This report explains the impact of these exploits, how they were resolved, next steps for the project, and our key learnings.

As painful as this process has been, it has hardened the team and the protocol. There is no doubt that the project is now stronger because of it.

The attack took advantage of several vulnerabilities:

1. Miner reward validation hack
2. xJPY to xBTC conversion/transfer
3. Hidden burn/mint amount bug
4. Zero value price record due to oracle being disabled

The fact these vulnerabilities were possible has highlighted weaknesses in our development processes. These have been fundamentally overhauled following recent events.

Once we are 100% confident that the protocol is secure and appropriate measures are in place, we will execute a hard fork. This will distribute an updated version of the code and potentially instigate a rollback.

It is worth noting that much of the analysis required during the recent investigation relies on the visibility of Haven Protocol's xAsset mint and burn data, which is intentionally public. Addresses and balances on Haven's chain remain private by default. We were also able to take advantage of the fact our anonymity set on xAssets is very small and not currently mixed sufficiently. This is a known privacy issue on xAssets that will also be addressed in the next fork.

Issue Summary

The recent investigations have been extensive, with both internal and external input, leading to the identification of several bugs and issues.

Blockchain scanner

Each issue has left an identifiable anomaly in the blockchain data. This meant it was possible to build a blockchain scanner, which scans the entire chain to form a complete list of affected transactions. This allows us to better understand the extent and impact of each exploit. It also gives us a high level of confidence that we have not missed any transactions.

The block scanner code: https://github.com/haven-protocol-org/haven-main/commit/8dae47791fe734e9081edc419c1ad91169b4b824

The block scanner results: https://docs.google.com/spreadsheets/d/1cP-rQtss9myY0j9_mbuWmdvyFwO_5xf30Kxwafd4NAQ/edit#gid=0

Notable entries in blockchain scan

To simplify the analysis, we've extracted the most important events in the table on the following page. These are transactions that resulted in an exploit, minting of assets, and inflation.

Date	Block / TX id	Flag	Extra Information	What is it?	How was it fixed?	Exploit Amount
6/22	882877	Mismatch in miner reward assets detected	Used assets = { XHV }, miner_tx claimed { XBTC XHV XUSD }	Miner reward exploit	Miner tx validation code fixed. added additional checks for validate_miner_transaction() · haven-protocol-org/haven-main@b9bd0e5 · GitHub	6.73 xBTC 101k xUSD
6/23	883040	Mismatch in miner reward assets detected	Used assets = { XHV }, miner_tx claimed { XBTC XHV XUSD }	Miner reward exploit	Miner tx validation code fixed. added additional checks for validate_miner_transaction() · haven-protocol-org/haven-main@b9bd0e5 · GitHub	6.73 xBTC 101k xUSD
6/24	884293	At least 1 input or 1 output of the tx was invalid	get_tx_type() failed	xJPY to xBTC - transfer bug	Fix to get_tx_asset_types function asset type bug fix · haven-protocol-org/haven-main@10931af · GitHub	total input 2.2 xJPY - xBTC NOT SPENT
6/24	884305	At least 1 input or 1 output of the tx was invalid	get_tx_type() failed	xJPY to xBTC - transfer bug	Fix to get_tx_asset_types function asset type bug fix · haven-protocol-org/haven-main@10931af · GitHub	Change of previous tx (4c87e7245142cb33a8ed4f039b7f33d4e4dd6b541a42a55992fd88efeefc40d1)
6/25	884689	At least 1 input or 1 output of the tx was invalid	get_tx_type() failed	xJPY to xBTC - transfer bug	Fix to get_tx_asset_types function asset type bug fix · haven-protocol-org/haven-main@10931af · GitHub	110 xJPY spent in ring member 10 of https://explorer.havenprotocol.org/tx/17f264dcfb1ebdc95e999668dcd6b2a1edbf32996e7337a564a2c65c3f47694d/1
6/27	886576	Missing conversion fee.	Source:XBTC, dest:XUSD, XHV fees:0,0, XUSD fees:0,0, burnt:0, minted:0	Perpetrators can hide burnt/mint amounts. No inflation caused but it can skew supply figures	In progress	Output value cannot be determined
6/29	887362	Missing conversion fee.	This is the first example, for a complete list see the full report	0 pricing record exploit	add 0 value check to daemon add 0 pr and amount bunt/mint check · haven-protocol-org/haven-main@7df47ce · GitHub	Output value cannot be determined

Exploit Analysis and Rollback

The only way to remove the effect of the exploits from the blockchain is to conduct a rollback. This section discusses the pros and cons of a rollback, and the optimum cut-off point.

What is a rollback?

A rollback is where the blockchain is cut, as part of a fork. Given that a blockchain is a ledger for all transactions, any transaction after the point where it is cut (rolled back) would be lost, and in effect reversed. It would be as if the ledger is rolled back in time. This method could be used to remove the impact of the hack.

This is an extreme response that would typically impact a large number of users, but in this case we have significantly mitigated the impact by closing deposits and withdrawals on exchanges to minimize the movement of funds on the chain.

Transactions within the exchange's closed system are managed on each exchange's database, not on the chain, so they should not be affected by a rollback.

How far back should the rollback go?

This is the key question. The block before any exploit could be considered a potential rollback point. In this case however, we have a number of exploits causing an increasing amount of inflation over a period of time. The further we roll back, the more hacks are removed, but the greater the disruption and potential loss for exchanges and users. The goal is to find a balance.

The optimal point is where an acceptable amount of inflation is removed, with an acceptable amount of disruption. Based on the data, and fact that exchanges closed on June 26th, there is only one point that meets these criteria. This is:

- Block: 886575
- Time: 2021-06-27 22:21:08

If we rollback to a block before this point, when the exchange wallets were open, users or exchanges could lose funds.

If we rollback to a block after this point, it would allow blocks with hidden mint burn data to survive, which hide unknown exploits.

What is the significance of exchange wallets?

We have considered rolling back further, to mitigate the impact of the earlier exploits. However, if we were to roll back to a point where exchange wallets were open, it would cause serious disruption.

The main impact to holders and exchanges is caused by deposits and withdrawals that occur during the rolled back period. Most exchanges were closed by block 886575, so rolling back to this point shouldn't affect the exchange's internal balance sheet.

Because of time differences, KuCoin was the last to close deposits and withdrawals. They actioned our request at 1:56 am on the June 28th. Block 886575 is only 3 hours and 35 minutes before this point, meaning we still have a short window for disruption. We hope to work with KuCoin to put this right.

If we were to roll back any further, far more deposits and withdrawals would be reversed, having a significant impact on exchanges and holders.

Inflationary Impact of a Rollback to Block 886575

If we roll back to 886575, all exploits before this point would be unaffected. We've calculated below the total inflationary impact of these exploits.

- Miner validation reward
  - Total exploit: 13.46 xBTC and 202,920 xUSD

- xJPY to xBTC conversion
  - Total exploit: 112.2 xBTC

- Grand Total: 125.66 xBTC and 202,920 xUSD

This is the total inflation that resulted from the two hacks, prior to the suggested rollback point. The exchange data we have seen so far suggests that most of these funds have already been sold and that the hacker does not hold much more, assuming we roll back to 886575 to wipe out later exploits.

It is critical to note that this total is also offset by the 440,000 XHV (~$1.5 million) that is currently frozen in suspected KuCoin accounts, and 100,000 XHV ($~350,000) that is frozen in suspected TradeOgre accounts. We hope to recover and burn these funds.

This would result in approximately $2.6 million in total inflationary impact, or 3% of Haven Protocol's current market cap, based on today's market prices. See the chart below for a summary:

Amount	USD Value	Status
125.66 xBTC	~$4,335,000	From exploit
202,920 xUSD	~$202,920	From exploit
440,000 XHV	~$1,540,000	Frozen in KuCoin
100,000 XHV	~$350,000	Frozen in TradeOgre

Community decision to rollback

It is important that all major decisions should be taken by the Haven Protocol community, including the decision to instigate a rollback.

Given that 886575 is the only logical rollback point, balancing acceptable results and manageable disruption, the decision for the community to answer is whether or not to rollback at all.

A vote will be held in Discord, as soon as possible. This will offer two choices:

1. Rollback chain to 886575
    - Pros: Will remove the largest exploits
    - Cons: Transactions after 886575 will be reversed

2. Do not rollback chain
    - Pros: No transactions will be reversed
    - Cons: Hackers will be left holding a large volume of XHV (Possibly 11m XHV), and supply figures will be unknown

Technical Documentation of Exploits

1. Miner reward validation

It was possible for an unscrupulous miner to modify the transaction code to exploit a vulnerability in the miner-reward-validation code. This meant that it was possible to mint a much higher mining reward than was due.

- Occurred: Blocks 882877 (2021-06-22 18:19:41) and 883040 (2021-06-23 00:01:50)
- Value of exploit: 2 equal transactions totaling 13.46 xBTC and 202,920 xUSD

Attempts were made to block these transactions in the patch to close the vulnerability. It is now unclear whether these attempts were successful.

Report

At 5 am on June 23, 2021 it came to our attention that there had been an attempted exploit on Haven Protocol. Upon investigation, we found that it was possible for an unscrupulous miner to modify the code to take advantage of a previously unknown vulnerability in the miner-reward-validation code. This meant that it was possible to mint a much higher mining reward than was due.

Because miner transactions are public, the abnormally high block reward meant that it was possible to identify which block contained the fraudulent transactions. As a result, we were able to analyze the chain and confirm that there were two instances where a miner had tampered with the code in an attempt to increase the block reward.

The developers were quickly able to replicate the exploit, design and implement a fix, and issue a patch that permanently prevents a future occurrence of this exploit.

Larger mining pools were first notified, who applied the patch around 1:30 pm the same day. As soon as a majority of nodes were running the patch, the exploit was disabled.

This patch was not made publicly available, as it would have drawn attention to the vulnerability before the remaining mining pools, and other solo miners had a chance to also update.

Follow-up

The team attempted to disable the counterfeit inputs in the deployed patch. We initially thought this was successful, but it’s now unknown if this attempt was successful or not.

Technical summary

Miner tx validation code has to validate each output of the miner tx. But the attacker was able to skip the validation on every other output by explicitly designing a malicious miner tx that allowed him to take advantage of a bug in the validation code.

2. xJPY to xBTC conversion

It was possible for the attacker to take advantage of a vulnerability in transaction types to modify outputs and mint extra xAssets.

- Occurred: 884293/2021-06-24 17:51:46 (2.2 xBTC), 884305/2021-06-24 18:09:30 (change from previous transaction), 884689/2021-06-25 07:04:19 (110 xBTC)
- Value of exploit: 2 transactions totaling 112.2 xBTC

Report

On the morning of June 28, 2021, the dev team became aware of two suspicious transactions in the explorer. A meeting was called immediately, and the team investigated the cause. After some initial exploration, we found a vulnerability that was exploited twice in three days, resulting in the minting of several counterfeit xBTC coins. This ultimately resulted in the unusually high selling volume of XHV on KuCoin.

Follow-up

Exchanges: We instructed exchanges to close wallets, to prevent anyone from profiting from the exploit, and to stabilize the situation.

Conversions: We disabled the pricing oracle to prevent any further conversions between XHV, xUSD, or other xAssets. This extreme measure was also necessary because the XHV price is invalid when the exchanges are closed.

Frozen accounts: KuCoin is supporting us in our investigation and actively freezing suspicious accounts.

Technical summary

Using an xJPY input and xBTC output, it was possible for the attacker to take advantage of a vulnerability in the get_tx_type() function. This function determines transaction type (offshore, onshore, etc.). By modifying outputs, they were able to make the transaction look like an xAsset transfer.

Since the transaction was being seen as a transfer rather than conversion, it bypassed the conversion validation code so the incorrect tx was accepted by the daemon.

3. *Hidden mint and burn value, vulnerability*

We found a bug that allowed the reporting of the actual number of assets minted or burnt to be manipulated. This isn't an exploit per se, and it doesn't allow for any inflation, but it does allow a bad actor to hide transactions.

It is possible to identify the transactions in the block scanner report, so we can see that it was used extensively after 886595.

This is why it is impossible to trust the supply figures after this block. We have seen 35 instances of this in the last scan, starting at block 886595.

4. *Zero value price record vulnerability*

The attacker was able to manipulate the output values to mint arbitrary amounts.

- Occurred: 18 times between 887361 (2021-06-29 00:45:20) and 887409 (2021-06-29 02:15:23)
- Value of exploit: It is not possible to determine the value of these exploits.

Report

In response to the xJPY exploit, the decision was taken to disable conversions by disabling pricing records. This was intended to block exploits in conversions and mitigate the impact of a rollback - if needed.

The protocol is designed to invalidate conversions when no price record is available. However, a vulnerability in this protocol meant that it was possible to exploit the zero price when constructing a transaction to mint additional funds.

Technical summary

The attacker was able to pass through proof-of-value and burnt/mint checks by modifying his tx and setting the amount burnt/mint to 0. Since 0 * x = 0, the validation passed, as it is supposed to return a 0 result. This is because inputs - (outputs + fee) should normally = 0.

This then allowed the attacker to manipulate the output values to mint an arbitrary amount.

Current Development Plan

Scheduled fork

Haven v1.4.0 hard fork was in progress before the exploits halted it. All vulnerabilities have now been addressed and will be rolled into this next fork.

Trading and conversions cannot resume until this fork is live. This fork will not be released until satisfactory penetration testing, third party reviews, and audits are complete. We will continue to keep the community updated on all progress.

Because exchange wallets are already closed, the usual two-week notice for a fork will not be required.

If the community requests a rollback, the logic to implement this will be included in this release.

V1.4.0 changelog

Original changes include:

xAsset Price lag changes

- Increase the lock time between xAsset conversions to 48 hours
- Increase xAsset conversion fee to 0.5%
- Implement 80% burn on xAsset conversion fee
- Split balance of xAsset conversion fee evenly between miner and governance wallets

Bug fixes and improvements

- Improve mixing of xAsset conversions (including database migration)
- Remove failed conversions from tx from the pool at point of failure - rather than 24 hours later (caused by tx pricing record height being older than ten blocks)
- Fix integer overflow bug on supply page - causing circulation discrepancies

Additional Vulnerability Patches

- Fix miner reward issues added additional checks for validate_miner_transaction · GitHub
- Fix xAsset conversion issues asset type bug fix · GitHub
- Fix 0 price record issue add 0 pr and amount bunt/mint check · GitHub
- Fix conversion fee overflow issue (found in block scan)
- Implement “proof-of-coin” into the protocol

Proof of coin

Haven Protocol validation is based on a proof of value. We are now extending this concept to include public mint and burn data, to ensure that it matches the hidden values in the proof of value calculation.

This gives the protocol a second layer of validation, ensuring any future attempt to manipulate mint and burn data will not be valid and cause the transaction to be rejected.

This will block all attack vectors related to the mint and burn exploits, like those we have seen recently.

This is a major improvement in the security of the protocol. An updated version of the white paper will be released to further explain this new validation.

Expected timeline

The team are working as fast as possible to make the network secure. However, this work cannot be rushed, and time will be taken where needed to ensure it is robust and properly checked.

Key milestones

- Build chain scanner - done
- Fix exploits - done
- Complete proof of coin - 70% done
- Testing (inc. 3rd party) - ongoing
- Fork/potential rollback/open exchanges and conversions - final steps

Lessons Learned from the Hack

In addition to the code updates discussed above, it is critical we learn from these experiences and fix the problems that have led to these issues. The key learnings are:

Improve development processes

- Open up the repository to more developers and ensure git history included. See github.
- Implement a master, develop, feature, and hotfix branch to make the process more open.
- We will maintain a standard of imposing unit tests that cover all edge cases before merging a feature branch into the development branch.
- Pull Requests will be transparent and reviewable by all. 2 members of the team must sign off on all PR's (2 of either Neil, Akil, or Justin).
- Rewrite Monero's unit tests for Haven.
  - Run these in a CI/CD process for every PR. Spend as much time as necessary reviewing every instance in the code where invalid inflation can feasibly be introduced.
- Add unit tests for each bullet below. Community members can aid us in adding tests, and we can develop an increasingly large list that is provably tested against, permanently included to run in the suite of tests that run every PR merge in the CI/CD pipeline.
  - Transaction creation.
  - Use modified conversion rates.
  - Convert XHV <> xAsset, xAsset <> different xAsset.
  - Incompatible transfer types
  - XHV <> xUSD, XHV <> xAsset, xUSD <> xAsset, xAsset <> different xAsset.
  - Multiple assets: XHV <> xUSD + xAsset, XHV <> xAsset + different xAsset, xUSD <> xAsset + different xAsset.
  - Utilize older fee versions from before xAssets and xUSD were introduced.
  - Utilize Monero's older tx versions to generate new output types.

    - Hard fork should probably simply prevent tx.version < 3.
    - Miner transaction.
    - Include minted coins of various assets, using various constructions with a keen eye on conditional logic.
    - Pricing record.
    - Arbitrary prices.
    - Use an earlier time stamped pricing record.
    - 0 values for any price.
    - 0 value for the signature and arbitrary prices.
- Scan the chain for any transactions or pricing records included which may have utilized any of the mechanisms above to create hidden inflation.
- A generous bug bounty program.
- Weekly or bi-weekly technical calls for anyone in the community to join and discuss technical ideas implemented and being considered for implementation in the future.
- Haven improvement proposals (HIPs).
- Repository to keep track of improvement proposals to Haven, and offer a forum for streamlined, transparent, asynchronous discussion.
- PR's will be managed by 2 people from the team.
- Implement a robust decentralized voting mechanism.
- Design and implement proof-of-coin to allow us to be confident the transparent amount minted and amount burnt on transactions is accurate going forward.
    - This will be reviewed, vetted, and validated by experts in cryptography, both the mathematical logic behind it, as well as the implementation.

Pool Operators

Mining rewards are currently 5,000 XHV per day. Pool operators stand to lose out if the rollback occurs.

Given pool operators' valuable contribution to the project, the team would be in favor of compensating them for lost mining rewards using funds from the governance wallet.

This is based on the assumption that the community are in favor of the idea and that the pools are able to calculate and distribute funds effectively.

Voting on Decisions

We've long known that as the decisions become more important, the voting process needs to improve. Over the long term, we aspire for a robust and fully decentralized mechanism to empower the Haven Protocol community and contributors.

That said, at this time we need something a little simpler, at least until a more robust system can be implemented.

We intend to allow the community to vote via a Discord poll. We appreciate that this method has weaknesses, but this is currently our only option given the time frame.

Measures will be taken to exclude and/or identify bots. We will also make sure that the community is properly informed before each vote.

Conclusion

We recognize that many of the mitigation actions detailed here are reflective of centralized protocols. As outlined in The Path Ahead this past April, our goal has and continues to be to move Haven towards an entirely decentralized future. We continue to believe we are 18–24 months from that state. In the meantime, we elected to make painful decisions at present that we felt protected new and old investors alike, that did not compromise Haven Protocol's ultimate mission, and that would provide the community the highest degree of confidence in the privacy and security of their holdings.

Our decision to engage law enforcement was also not taken lightly given the protocol's privacy focus, and we attempted to ensure the safety of the XHV community without it. However, this formal involvement is mandated by our exchange partners in order to permanently freeze the accounts that continue to hold a substantial amount of exploited XHV.

We are deeply apologetic for the pain and anxiety these events have caused so many in our community. We proudly took over development of Haven in early 2019 after the prior development team had obfuscated their progress, obviated their duties, and abandoned the same community they claimed to serve. We remain committed to Haven as one of the most important projects and technologies in the entire crypto ecosystem. In spite of our substantial progress these past 24 months, we recently rushed the testing process to increase the speed of xAsset progress. This decision ultimately had a detrimental effect and we are deeply committed to shoring up our code and keeping your assets safe, private, and stable henceforth.

Thank you again to our incredible community for their support, patience, and assistance during these challenging weeks. Without you, there would be no Haven Protocol. We know it has not been easy given the uncertainty of the situation. We are committed to taking every necessary step to protect and strengthen the project in the days, weeks, and months ahead.