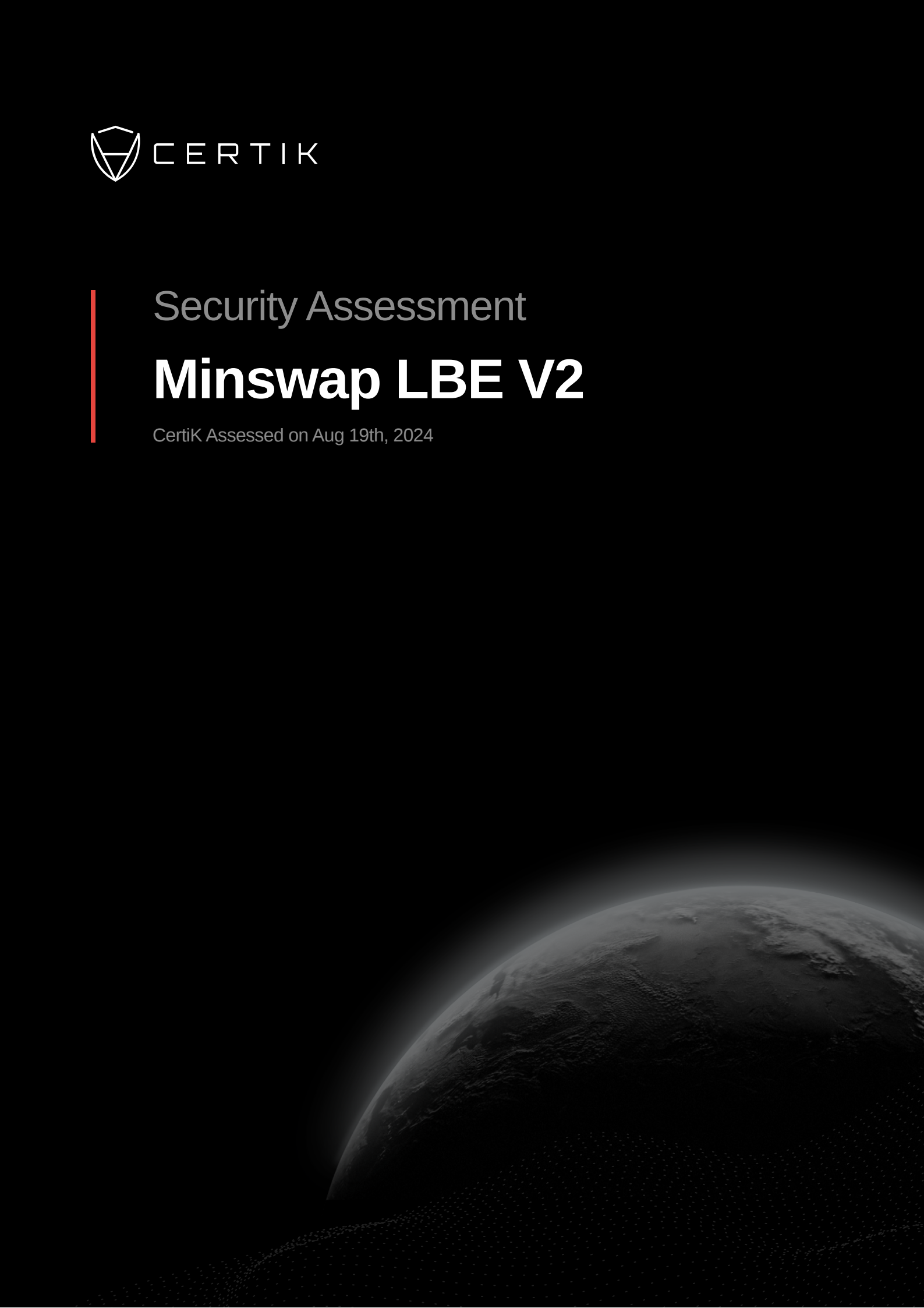
CERTIK

Security Assessment

Minswap LBE V2

CertiK Assessed on Aug 19th, 2024



CertiK Assessed on Aug 19th, 2024

Minswap LBE V2

The security assessment was prepared by CertiK, the leader in Web3.0 security.

Executive Summary

TYPES	ECOSYSTEM	METHODS
Exchange	Cardano (ADA)	Manual Review, Static Analysis

LANGUAGE	TIMELINE	KEY COMPONENTS
Aiken	Delivered on 08/19/2024	N/A

CODEBASE
https://github.com/minswap/minswap-lbe-v2/
View All in Codebase Page

COMMITS
8ab54ca1df99c84e3a449d2d1f5d9f4ed7c77a76
94f1c742904417b053f2160747232f474bb37aa0
View All in Codebase Page

Vulnerability Summary

4	2	0	0	2	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined

0	Critical		Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
2	Major	1 Resolved, 1 Acknowledged	Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
0	Medium		Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
0	Minor		Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.
2	Informational	1 Resolved, 1 Acknowledged	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | MINSWAP LBE V2

CODEBASE | MINSWAP LBE V2

Repository

https://github.com/minswap/minswap-lbe-v2/

Commit

8ab54ca1df99c84e3a449d2d1f5d9f4ed7c77a76

94f1c742904417b053f2160747232f474bb37aa0

AUDIT SCOPE | MINSWAP LBE V2

12 files audited ● 4 files with Acknowledged findings ● 1 file with Resolved findings ● 7 files without findings

ID	Repo	File	SHA256 Checksum
CKP	CertiKProject/certik-audit-projects	lib/lb_v2/cancel_validation.ak	a508057f20f07db64b66bb4624cde55e6cafa3079dde792760e77497ae883ff7
ORD	CertiKProject/certik-audit-projects	lib/lb_v2/order_validation.ak	e15ddde02bcf887d83af37fb7437b70da20fa79f38947f1d888206e9881d8d8f
FAC	CertiKProject/certik-audit-projects	validators/factory.ak	ab07394aa4dcab92f49efa7f3720a7f6a9ca6b2b01798fdde4841b2a03980d88
TRA	CertiKProject/certik-audit-projects	validators/treasury.ak	f958170f61793a986b002960c55efe949a5ef64b4794678a49df0a3c52995308
VAI	CertiKProject/certik-audit-projects	lib/lb_v2/validation.ak	4930f7eb08bb7fefbf7cd0adf0008696f1b1e2fdf9a7e89d1c805f1ab2248e0c
MAN	CertiKProject/certik-audit-projects	lib/lb_v2/manager_validation.ak	b95eed1b7f04b7089837614daaf6eef3897422e005b79ccadaec47f24aad58e0
TRE	CertiKProject/certik-audit-projects	lib/lb_v2/treasury_validation.ak	8ccaf652b0332dc00b6a20bee4abd301a02f1e9b9e081de5f29a9da8e89133bb
TYP	CertiKProject/certik-audit-projects	lib/lb_v2/types.ak	88dfca575edd54bf00301820faea61f6567c0e8285f4695a6e70b625c0cf6988
UTI	CertiKProject/certik-audit-projects	lib/lb_v2/utils.ak	64e6529a9c2def181f956009b166fb3f4723579e2dc8f712c9385f6e458cbb84
MAA	CertiKProject/certik-audit-projects	validators/manager.ak	ff3a58f4f8d14a69ed6a79d84d1e1abde3cb9eb9c11bc50d06e7602f34a30162
ORE	CertiKProject/certik-audit-projects	validators/order.ak	840b28a72b20921c17296f84737b773516a8c68def7707eb1376456a2ebb28e7
SEL	CertiKProject/certik-audit-projects	validators/seller.ak	efae25b4de6af91de74bbefecce5fb81bcba2777386d58481eab64830efd0887

APPROACH & METHODS | MINSWAP LBE V2

This report has been prepared for Minswap to discover issues and vulnerabilities in the source code of the Minswap LBE V2 project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review and Static Analysis techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors;
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

FINDINGS | MINSWAP LBE V2

4	0	2	0	0	2
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Minswap LBE V2. Through this audit, we have uncovered 4 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
GLOBAL-01	Missing Check On The Authorization Of Creating LBE	Inconsistency	Major	Resolved
MIN-01	**Centralization Related Risks**	**Centralization**	**Major**	**Acknowledged**
FAC-02	Comment Error	Inconsistency	Informational	Resolved
ORD-01	Potential Unchecked Condition	Logical Issue	Informational	Acknowledged

GLOBAL-01 MISSING CHECK ON THE AUTHORIZATION OF CREATING LBE

Category	Severity	Location	Status
Inconsistency	Major		Resolved

Description

According to the whitepaper, the creation of a new LBE should be initiated by the project owner. However, the current implementation lacks validation to ensure that only the project owner can create a new LBE and mint Factory, Treasury, Manager, & Seller tokens. This vulnerability can be exploited by malicious actors to create unauthorized LBEs and mint aforementioned tokens, potentially leading to significant financial loss and reputation damage.

```
134         CreateTreasury { .. } -> {
135           let Transaction { datums, .. } = transaction
136           // finding exactly 1 Factory Input
137           expect [factory_input] = factory_inputs
138           // finding exactly 2 Factory Outputs
139           expect [f_out_head, f_out_tail] = factory_outputs
140           // finding exactly 1 Treasury Input
141           let treasury_output =
142             validation.get_treasury_output(
143               outputs: outputs,
144               factory_policy_id: factory_hash,
145               treasury_hash: treasury_hash,
146             )
147           // finding exactly 1 Manager Output
148           expect [manager_output] =
149             list.filter(
150               outputs,
151               fn(output) {
152                 let Output { address: Address { payment_credential, .. }, .. }
=
153                   output
154                 // output belongs Manager Address
155                 payment_credential == ScriptCredential(manager_hash)
156               },
157             )
158           // Extract some necessary data.
159           let Input { output: Output { value: factory_input_value, .. }, .. } =
160             factory_input
161           let Output {
162             datum: f_out_head_datum,
163             value: f_out_head_value,
164             reference_script: f_out_head_ref_script,
165             ..
166           } = f_out_head
167           let Output {
168             datum: f_out_tail_datum,
169             value: f_out_tail_value,
170             reference_script: f_out_tail_ref_script,
171             ..
172           } = f_out_tail
173           let mint_value = value.from_minted_value(mint)
174           let mint_seller_count =
175             value.quantity_of(mint_value, factory_hash, seller_auth_an)
176           let default_manager_output =
177            build_default_manager_output(
178               factory_policy_id: factory_hash,
179               manager_hash: manager_hash,
180               base_asset: base_asset,
181               raise_asset: raise_asset,
182               seller_count: mint_seller_count,
183             )
184           // Trivial Assertions
```

```
185          expect
186            assert(
187              value.quantity_of(
188                factory_input_value,
189                factory_hash,
190                factory_auth_an,
191              ) == 1,
192              @"Factory Input must be Legit!",
193            )
194          expect assert(and {
195                // validate that new Factory UTxO datum must be followed by
Linked List rule
196                // (old head, old tail) -> (old head, LP Token Name) and (LP
Token Name, old tail)
197                // old head < LP Token Name < old tail
198                builtin.less_than_bytearray(current_head, lp_asset_name),
199                builtin.less_than_bytearray(lp_asset_name, current_tail),
200                // Factory Output must contains 1 Factory Token
201                value.quantity_of(f_out_head_value, factory_hash,
factory_auth_an) == 1,
202                value.quantity_of(f_out_tail_value, factory_hash,
factory_auth_an) == 1,
203                // Factory Output must contains only ADA and Factory Token
204                list.length(value.flatten(f_out_head_value)) == 2,
205                list.length(value.flatten(f_out_tail_value)) == 2,
206                // Head Factory Datum must be correct!
207                f_out_head_datum == InlineDatum(
208                  FactoryDatum { head: current_head, tail: lp_asset_name },
209                ),
210                // Tail Factory Datum must be correct!
211                f_out_tail_datum == InlineDatum(
212                  FactoryDatum { head: lp_asset_name, tail: current_tail },
213                ),
214                // Prevent Factory Output becoming heavy!
215                f_out_head_ref_script == None,
216                f_out_tail_ref_script == None,
217              }, @"2 Factory Outputs must pay correctly!")
218          // Assertions:
219          and {
220            // Manager Output must pay correctly!
221            manager_output == default_manager_output,
222            // Must prepare enough Sellers
223            mint_seller_count >= minimum_number_seller,
224            // Seller Outputs must pay correctly!
225            validation.validate_seller_outputs(
226              outputs: outputs,
227              factory_policy_id: factory_hash,
228              base_asset: base_asset,
229              raise_asset: raise_asset,
230              seller_hash: seller_hash,
231              seller_count: mint_seller_count,
232            ),
233            // Treasury Output must pay correctly!
```

```
234            validate_creating_treasury_out(
235              treasury_out: treasury_output,
236              base_asset: base_asset,
237              raise_asset: raise_asset,
238              manager_hash: manager_hash,
239              seller_hash: seller_hash,
240              order_hash: order_hash,
241              factory_policy_id: factory_hash,
242              end_valid_time_range: end_valid_time_range,
243              datums: datums,
244            ),
245            // Mint Value must be correct!
246            mint_value == get_minting_treasury(
247              factory_policy_id: factory_hash,
248              seller_count: mint_seller_count,
249            ),
250          }
251        }
```

Recommendation

We recommend the team ensuring the implementation consistent with the design.

Alleviation

[Minswap Team, 2024/08/09]: We fixed this issue in PR https://github.com/minswap/minswap-lbe-v2/pull/47#pullrequestreview-2217007379. However, there is a feature where the Treasury Wallet (holding the Project's Token) and the Project's Owner Wallet are separate. In a specific scenario:

Treasury Wallet: Holds the project's token, initialized by the TGE (Token Generation Event). This could be a PubKey Wallet, Multi-Sig Wallet, or even a smart contract.

Project's Owner Wallet: Controlled by the DAO. This can be a PubKey Wallet, a Multi-Sig Wallet, or a smart contract.

[CertiK, 2024/08/09]: The team heeded the advice and resolved the finding in commit f7584f86730169cfb7932636e79a8700c41f943a.

MIN-01 | CENTRALIZATION RELATED RISKS

Category	Severity	Location	Status
Centralization	**Major**	**lib/lb_v2/cancel_validation.ak: 84~109; validators/factory.ak: 347~351; validators/treasury.ak: 110~114**	**Acknowledged**

Description

In the `cancel_validation.ak`, the project owner can cancel a LBE if needed before discovery phase starts or cancel a LBE before discovery phase ended when `revocable` is true. Any compromise to the project owner account may allow a hacker to take advantage of this authority and cancel all qualified LBEs.

In the `factory.ak`, the project owner can close a LBE. Any compromise to the project owner account may allow a hacker to take advantage of this authority and delay the closing of LBEs.

In the `treasury.ak`, the project owner can update uncanceled LBE parameter fields except for the `base_asset` and `raise_asset` field before the start of the discovery phase. Any compromise to the project owner account may allow a hacker to take advantage of this authority and updating those parameters arbitrarily.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multisignature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND

- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles.
  OR
- Remove the risky functionality.

Alleviation

[Minswap Team, 2024/08/09]: "Issue acknowledged. I won't make any changes for the current version. In the Permission-less LBE protocol, an important feature is the support for various types of credentials: public/private key pairs or a script (Native or Plutus). To further enhance security and reduce the risk associated with a single point of failure in key management, projects have the option to use a script as the Project's Owner actor. This approach provides an additional layer of protection."

[CertiK, 2024/08/09]: It is suggested to implement the aforementioned methods to avoid centralized failure. Also, CertiK strongly encourages the project team to periodically revisit the private key security management of all addresses related to centralized roles.

[Minswap Team, 2024/08/19]: "We acknowledge Certik's concern about the potential risk of centralization failure if a Project Owner loses their private key, leading to a possible compromise where an attacker could cancel the LBE event and steal the project tokens. However, it's important to note that this risk does not originate from Minswap itself but is tied to the security practices of the Project Owner. In any case, if an LBE event is cancelled, the protocol ensures that all user funds are safely returned to participants, so user funds are not at risk.

Certik recommended that only multi-signature wallets or smart contracts be allowed to initiate LBE events to mitigate this risk. While this would indeed strengthen security, it could also create barriers for many projects, particularly those run by teams without the technical capability to set up such wallets or contracts.

Given that the LBE protocol is permissionless and designed to support normal wallets, multi-signature wallets and smart contracts. it’s essential to maintain this flexibility. This approach allows us to serve a wide range of projects, ensuring the protocol remains accessible to both technical and non-technical teams.

To strike a balance, we will continue to allow both normal wallets (with private keys) and multi-signature wallets to create LBE events. Additionally, for projects seeking greater security and no risk of centralization failure, we will encourage and educate about the use of multi-sign and Smart contracts which can provide an extra layer of protection and encourage the project team to periodically revisit the private key security management of all addresses related to centralized roles.

This solution upholds the protocol’s accessibility while offering enhanced security options for those who need them."

FAC-02 | COMMENT ERROR

Category	Severity	Location	Status
Inconsistency	Informational	validators/factory.ak: 140	Resolved

Description

When validating a treasury creation in `factory`, the following comment:

```
140          // finding exactly 1 Treasury Input
141          let treasury_output =
142            validation.get_treasury_output(
```

describe a check on the input but only outputs are checked.

Recommendation

We recommend rewriting the comment as follows

```
        // finding exactly 1 Treasury Output
```

Alleviation

[Minswap Team, 2024/08/09]: This issue has been fixed in PR https://github.com/minswap/minswap-lbe-v2/pull/46

ORD-01 | POTENTIAL UNCHECKED CONDITION

Category	Severity	Location	Status
Logical Issue	Informational	lib/lb_v2/order_validation.ak: 90~96	Acknowledged

Description

The function `validate_collect_orders` in `order_validation.ak` makes an assumption that the orders collecting in the last round must be smaller than `minimum_order_collected`. Additionally, due to Plutus' short circuit evaluation, the second check `collected_fund + collect_amount == reserve_raise + total_penalty` may not get validated at all.

```
90      or {
91        // prevent spamming by setting minimum for orders collected
92        //  if this tx is not the last collect action
93        list.length(order_inputs) >= minimum_order_collected,
94        // the last collecting
95        collected_fund + collect_amount == reserve_raise + total_penalty,
96      },
```

Recommendation

We recommend the team confirming if it's the intended design.

Alleviation

[Minswap Team, 2024/08/09]: The "Collect Orders" transaction can handle up to 50 orders per batch. If there are 31 orders left, the last batch will process these 31 orders, so the second condition won't be checked. But if there are only 10 orders left, the second condition will be triggered and allow the collection of these 10 orders.

[CertiK, 2024/08/09]: The team confirmed that it's an intended design and there's a possibility that the second condition won't be checked. The `collected_fund + collect_amount == reserve_raise + total_penalty` will still be checked in the `validate_create_dex_pool`.

OPTIMIZATIONS | MINSWAP LBE V2

ID	Title	Category	Severity	Status
FAC-01	`assert` Statements Inside `and` Block	Code Optimization	Optimization	Resolved
VAI-01	Redundant Datum Validation	Code Optimization	Optimization	Resolved

FAC-01 | `assert` STATEMENTS INSIDE `and` BLOCK

Category	Severity	Location	Status
Code Optimization	Optimization	validators/factory.ak: 352~358, 511~537	Resolved

Description

In the `factory.validate_factory`, multiple `assert` statements are placed inside an `and` block. This approach is not optimized because `assert` should fail immediately if the condition is false, while `and` is designed to return a boolean value.

```
352            assert(is_cancelled, @"LBE should already cancelled"),
353            assert(
354              is_manager_collected,
355              @"All Manager, Sellers must be collected!",
356            ),
357            assert(reserve_raise == 0, @"All Orders have been executed."),
358            assert(total_penalty == 0, @"All Penalty have been handled."),
```

A similar case can be found in `factory.validate_initialization`.

Recommendation

We recommend refactoring the validator logic to place `assert` statements before the `and` block in the execution flow. This will optimize the validation process by catching issues sooner and reducing unnecessary computations.

Alleviation

[Minswap Team, 2024/08/09]: This issue has been fixed in PR https://github.com/minswap/minswap-lbe-v2/pull/46

VAI-01 | REDUNDANT DATUM VALIDATION

Category	Severity	Location	Status
Code Optimization	Optimization	lib/lb_v2/validation.ak: 596~603, 610~617	Resolved

Description

In the function `validation.validate_seller_outputs`, the following piece of code:

```
610   expect
611     datum == SellerDatum {
612       factory_policy_id,
613       base_asset,
614       raise_asset,
615       amount: 0,
616       penalty_amount: 0,
617       owner,
618     }
```

will stop the execution if the datum is incorrect. The same check is made afterward to make the function return false if the datum is not correct. This redundancy can lead to unnecessary code execution and potential performance issues.

Recommendation

We recommend removing the redundant validation check to streamline the function and improve performance.

Alleviation

[Minswap Team, 2024/08/09]: This issue has been fixed in PR https://github.com/minswap/minswap-lbe-v2/pull/46

APPENDIX | MINSWAP LBE V2

Finding Categories

Categories	Description
Inconsistency	Inconsistency findings refer to different parts of code that are not consistent or code that does not behave according to its specification.
Logical Issue	Logical Issue findings indicate general implementation issues related to the program logic.
Centralization	Centralization findings detail the design choices of designating privileged roles or other centralized controls over the code.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

DISCLAIMER | CERTIK

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR

UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

CertiK | Securing the Web3 World

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.