CERTIK

Security Assessment

Diment - (Main Contract & Proxy)

CertiK Assessed on Mar 27th, 2024

CertiK Assessed on Mar 27th, 2024

Diment - (Main Contract & Proxy)

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

Executive Summary

TYPES	ECOSYSTEM	METHODS
ERC-20	Ethereum (ETH)	Manual Review, Static Analysis
LANGUAGE	**TIMELINE**	**KEY COMPONENTS**
Solidity	Delivered on 03/27/2024	N/A

CODEBASE

0x522ccdc13c63fa7372cd191f9f4facfa8b288b88
0xf42583b9f731a752008552a31a396f53c8615e5e
0x7e10e925c2bb624b580507d75Dc9DBAB7f3DE055
View All in Codebase Page

COMMITS

67904b3228060821d9335646bfadc8c134a64d36 /
1d1527209092cdd04b3576a4a5f583281a57589b /
f0d9012e20259e5654fe7a1c2d77b030b88fcc70
View All in Codebase Page

Vulnerability Summary

4	1	3	0	0	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.
3	Major	3 Mitigated	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.
1	Minor	1 Resolved	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.
0	Informational		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 | DIMENT - (MAIN CONTRACT & PROXY)

CODEBASE | DIMENT - (MAIN CONTRACT & PROXY)

Repository

0x522ccdc13c63fa7372cd191f9f4facfa8b288b88 0xf42583b9f731a752008552a31a396f53c8615e5e
0x7e10e925c2bb624b580507d75Dc9DBAB7f3DE055 0xcE64B554a10910BAC66Beb70bdfc87CE655efb9B
0xA05cf964B41667ccda963E147C86b4C2E2499C33

Commit

67904b3228060821d9335646bfadc8c134a64d36 / 1d1527209092cdd04b3576a4a5f583281a57589b /
f0d9012e20259e5654fe7a1c2d77b030b88fcc70
d2a84ae26e4767889a2d720aea00d821321873b2
594bb5d5197914fbd8655f272154050c3d73be6c

AUDIT SCOPE | DIMENT - (MAIN CONTRACT & PROXY)

44 files audited ● 2 files with Mitigated findings ● 42 files without findings

ID	Repo	File	SHA256 Checksum
DDH	goerli	contracts/DimentDollar.sol	2f6ae15b021cea98ad153910c27bc7ed48f368aaf10c2bdadee73f6de237d51a
TUP	goerli	@openzeppelin/contracts/proxy/transparent/TransparentUpgradeableProxy.sol	eca2efb275f85f4440db6d4dd2086be0700082e6caea8f5995b7da0446a622f6
OUH	goerli	@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol	1cd6a7cd8e2270eb039210ccff54fa50f8ad2932e22820cc267749cf4f9b16b4
INI	goerli	@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol	298ba69c2c51f74db09f2451edf2f7bfaf3092542a48a9fc9b457a2f6e7e35e8
ERP	goerli	@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC20PermitUpgradeable.sol	ece9cfa50737ed7ecb2d096572af145ca6a2a18ce9da3acc365e73ce42a726fa
ERU	goerli	@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol	4932bb558ec19e77f0aa804dd09f38bfefae63171f7dcbbfc3edfa86a4e41fbf
EIP	goerli	@openzeppelin/contracts-upgradeable/utils/cryptography/EIP712Upgradeable.sol	9db2d9b07f1483cdc73689052cce583539b22dfac75e2fe21a528749111b432b
CUH	goerli	@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol	5da1fd51dbcc63e29bd869bf32880bb3f2fa7de5c9226658b7b03e27b42ce72b
NUH	goerli	@openzeppelin/contracts-upgradeable/utils/NoncesUpgradeable.sol	83613b3c450d824ba4f03dde03ba2e79a765b8a4b1929a3f95a516bcbfd0f0c2
IER	goerli	@openzeppelin/contracts/interfaces/IERC5267.sol	87936cc2ceaf511f743797a50be8e406c0239e8b970ee1d059579b0de2f6b782
IEC	goerli	@openzeppelin/contracts/interfaces/draft-IERC6093.sol	5339c9008dd0d5a288e1514d67d47ba7cd0a1b3fca38a2c8994dadb875bce737

ID	Repo	File	SHA256 Checksum
IEM	goerli	@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	47b68f3cbc09a04e245dbd4c88a37f7cc1b523d226311b610e458530c7133fab
IEP	goerli	@openzeppelin/contracts/token/ERC20/extensions/IERC20Permit.sol	e16c38a896f4fdf834d52f511d9f8f0e1817a33b128fbaa52b742897a52058a1
IEE	goerli	@openzeppelin/contracts/token/ERC20/IERC20.sol	101c1119eaec82febe17e8b792791580761a2bf182d0d2d2e04a1cf29a2e09d4
ECD	goerli	@openzeppelin/contracts/utils/cryptography/ECDSA.sol	4390642fd68a12485b7b334fc768a35707510a962ffdaf5bfaf8272416200c4b
MHU	goerli	@openzeppelin/contracts/utils/cryptography/MessageHashUtils.sol	d23627291b30276cbc0e962e7648e335317d27c502db55902028ac8004f67770
MAH	goerli	@openzeppelin/contracts/utils/math/Math.sol	0a5e8697c5e155214368b95a212b4a2db44c73a4e4ab151ebea7eb189eb4cc18
SMH	goerli	@openzeppelin/contracts/utils/math/SignedMath.sol	99e525c92b7da36bab7fcee2838fd74069f7f71524a6c0546cab8d41fc9f434e
STR	goerli	@openzeppelin/contracts/utils/Strings.sol	1362240e8812d8556eae5b1aeedb7c0987a74d36bdd46dc7f4e9f6f4135f0cce
OWN	goerli	@openzeppelin/contracts/access/Ownable.sol	38578bd71c0a909840e67202db527cc6b4e6b437e0f39f0c909da32c1e30cb81
IRC	goerli	@openzeppelin/contracts/interfaces/IERC1967.sol	886b093d8f7c41f73af42b8e183314b3654531a9d5e11f07c41a5a7f11d3e006
ERE	goerli	@openzeppelin/contracts/proxy/ERC1967/ERC1967Proxy.sol	ca1c1476f97761f3a5830395576c82756899ad8896489cef9c388afab825ef21
ERR	goerli	@openzeppelin/contracts/proxy/ERC1967/ERC1967Utils.sol	8850e97f15234cf93d7d1828b6289aeda7fa7167b3550b2f2a9713c8e2cecc80
BPH	goerli	@openzeppelin/contracts/proxy/beacon/BeaconProxy.sol	12873ed28845bbc2c3a17926c17ed8b36fac3d3b22f53a2f3dd7beaff1c721a6
IBH	goerli	@openzeppelin/contracts/proxy/beacon/IBeacon.sol	422eabc0e645e24c3a52898f6255b349323b013544a3ebdc4b2d3f7fc5bb7e9e
UBH	goerli	@openzeppelin/contracts/proxy/beacon/UpgradeableBeacon.sol	26dda9d5bb961b3df26602d49f9f5a0647cfdb78b63cc253aed8527030a64f25

ID	Repo	File	SHA256 Checksum
PAH	goerli	@openzeppelin/contracts/proxy/transparent/ProxyAdmin.sol	29419f1bd5a3ca58870e7aee3bb2b658f1f31986975844c8f09146179985778b
PRY	goerli	@openzeppelin/contracts/proxy/Proxy.sol	5f5081378d4bc82b814b0d64990b7f7b9c69697593b73a3341f4a269940ba540
ADD	goerli	@openzeppelin/contracts/utils/Address.sol	b3710b1712637eb8c0df81912da3450da6ff67b0b3ed18146b033ed15b1aa3b9
COE	goerli	@openzeppelin/contracts/utils/Context.sol	847fda5460fee70f56f4200f59b82ae622bb03c79c77e67af010e31b7e2cc5b6
SSH	goerli	@openzeppelin/contracts/utils/StorageSlot.sol	b4a5fb7ab93bfeda06509eafbd5f71fde0e0de84b6d9129553bd535a42166c15
OWA	goerli	@openzeppelin/contracts/access/Ownable.sol	38578bd71c0a909840e67202db527cc6b4e6b437e0f39f0c909da32c1e30cb81
IE1	goerli	@openzeppelin/contracts/interfaces/IERC1967.sol	886b093d8f7c41f73af42b8e183314b3654531a9d5e11f07c41a5a7f11d3e006
ECP	goerli	@openzeppelin/contracts/proxy/ERC1967/ERC1967Proxy.sol	ca1c1476f97761f3a5830395576c82756899ad8896489cef9c388afab825ef21
ECU	goerli	@openzeppelin/contracts/proxy/ERC1967/ERC1967Utils.sol	8850e97f15234cf93d7d1828b6289aeda7fa7167b3550b2f2a9713c8e2cecc80
BPT	goerli	@openzeppelin/contracts/proxy/beacon/BeaconProxy.sol	12873ed28845bbc2c3a17926c17ed8b36fac3d3b22f53a2f3dd7beaff1c721a6
IBT	goerli	@openzeppelin/contracts/proxy/beacon/IBeacon.sol	422eabc0e645e24c3a52898f6255b349323b013544a3ebdc4b2d3f7fc5bb7e9e
UBT	goerli	@openzeppelin/contracts/proxy/beacon/UpgradeableBeacon.sol	26dda9d5bb961b3df26602d49f9f5a0647cfdb78b63cc253aed8527030a64f25
PAT	goerli	@openzeppelin/contracts/proxy/transparent/ProxyAdmin.sol	29419f1bd5a3ca58870e7aee3bb2b658f1f31986975844c8f09146179985778b
TRS	goerli	@openzeppelin/contracts/proxy/transparent/TransparentUpgradeableProxy.sol	eca2efb275f85f4440db6d4dd2086be0700082e6caea8f5995b7da0446a622f6
PRP	goerli	@openzeppelin/contracts/proxy/Proxy.sol	5f5081378d4bc82b814b0d64990b7f7b9c69697593b73a3341f4a269940ba540

ID	Repo	File	SHA256 Checksum
ADR	goerli	@openzeppelin/contracts/utils/Address.sol	b3710b1712637eb8c0df81912da3450da6ff67b0b3ed18146b033ed15b1aa3b9
COX	goerli	@openzeppelin/contracts/utils/Context.sol	847fda5460fee70f56f4200f59b82ae622bb03c79c77e67af010e31b7e2cc5b6
SST	goerli	@openzeppelin/contracts/utils/StorageSlot.sol	b4a5fb7ab93bfeda06509eafbd5f71fde0e0de84b6d9129553bd535a42166c15

APPROACH & METHODS | DIMENT - (MAIN CONTRACT & PROXY)

This report has been prepared for Diment to discover issues and vulnerabilities in the source code of the Diment - (Main Contract & Proxy) 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 | DIMENT - (MAIN CONTRACT & PROXY)

4	0	3	0	1	0
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Diment - (Main Contract & Proxy). 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
DDH-01	**Centralized Balance Manipulation**	**Centralization**	**Major**	**Mitigated**
DDH-04	**Centralization Related Risks**	**Centralization**	**Major**	**Mitigated**
GOE-02	**Centralized Control Of Contract Upgrade**	**Centralization**	**Major**	**Mitigated**
DDH-02	Unprotected Initializer	Coding Issue	Minor	Resolved

DDH-01 | CENTRALIZED BALANCE MANIPULATION

Category	Severity	Location	Status
Centralization	**Major**	**contracts/DimentDollar.sol (DimentDollar): 151**	**Mitigated**

Description

In the contract `DimentDollar`, the role `onlyOwner` has the authority to update the token balance of an arbitrary account without enough sanity restriction.

Any compromise to the `onlyOwner` account may allow a hacker to take advantage of this authority and manipulate users' balances by either:

- minting any amount of token to any address;
- burning all tokens from a specific address by adding it to the blacklist;

Recommendation

We recommend the team makes efforts to restrict access to the private key of the privileged account. A strategy of multi-signature (⅔, ⅗) wallet can be used to prevent a single point of failure due to a private key compromise. In addition, the team should be transparent and notify the community in advance whenever they plan to mint more tokens or engage in similar balance-related operations.
Here are some feasible short-term and long-term suggestions that would mitigate the potential risk to a different level and suggestions that would permanently *fully* resolve the risk:

Short Term:

A multi signature (⅔, ⅗) wallet *mitigate* the risk by avoiding a single point of key management failure.

- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to a private key compromised;
  AND
- A medium/blog link for sharing the time-lock contract and multi-signers' addresses information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.
- Provide a link to the **medium/blog** with all of the above information included.

Long Term:

A DAO for controlling the operation *mitigate* the risk by applying transparency and decentralization.

- Introduction of a DAO, governance, or voting module to increase decentralization, transparency, and user involvement;
  AND
- A medium/blog link for sharing the multi-signers' addresses, and DAO information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.
- Provide a link to the **medium/blog** with all of the above information included.

Permanent:

The following actions can *fully* resolve the risk:

- Renounce the ownership and never claim back the privileged role.
  OR
- Remove the risky functionality.
  OR
- Add minting logic (such as a vesting schedule) to the contract instead of allowing the owner account to call the sensitive function directly.

Note: we recommend the project team consider the long-term solution or the permanent solution. The project team shall make a decision based on the current state of their project, timeline, and project resources.

Alleviation

[Diment team, 2024/02/28]: Issue acknowledged. I won't make any changes to the current version. We are a centralized token so we need these functions in our contract.

We will give ownership to a Multi-Signature Wallet contract whose implementation is included in the scope of this audit so hack risk is going minimum for mint and burn.

[CertiK, 2024/03/27]: 2024/03/27, 03:01:34 UTC, block 37323868 on Binance smart chain:

`DimentDollar` contract address is :

- **implementation**: 0xc545eed89bb404abbe5cfbf7c96643bcf4561309;
- **proxy**: 0x71b3a0566f4bf80331d115d8026a7022bf670cce;

The ownership of the **proxy** has been transferred, in this *transaction* to the `DimentTimelockController` deployed at address: 0xCfA0E2641Ce128959EbCDF680073E0C480B98442:

- its `DEFAULT_ADMIN_ROLE` is granted to addresses:
  - `DimentTimelockController`;
  - `DimentMultiSignatureWallet`;
- its `PROPOSER_ROLE` is granted to `DimentMultiSignatureWallet`;
- its `CANCELER_ROLE` is granted to `DimentMultiSignatureWallet`;
- its `EXECUTOR_ROLE` is granted to the signers of the `DimentMultiSignatureWallet`:
  1. 0xEb098A67D7c46cA48c701cd09d6A3A37b1BA0717 an EOA;
  2. 0x5D3C96bF7eCf9bDB75F18BEF5f4a7AEF351543Ea an EOA;
  3. 0xD5aE52e39750c52c94A725D7b7f717239d964AF5 an EOA;

the minimum delay is set to 48 hours.

The `DimentMultiSignatureWallet` has been deployed in this *transaction* with the following addresses as `owners_`:

1. 0xEb098A67D7c46cA48c701cd09d6A3A37b1BA0717 an EOA;
2. 0x5D3C96bF7eCf9bDB75F18BEF5f4a7AEF351543Ea an EOA;
3. 0xD5aE52e39750c52c94A725D7b7f717239d964AF5 an EOA;

`numConfirmationsRequired` is equal to 2.

The Diment team has applied the timelock with 48 hours, and the ⅔ multisig as a short-term solution. While this strategy has indeed reduced the risk, it's crucial to note that it has not completely eliminated it.

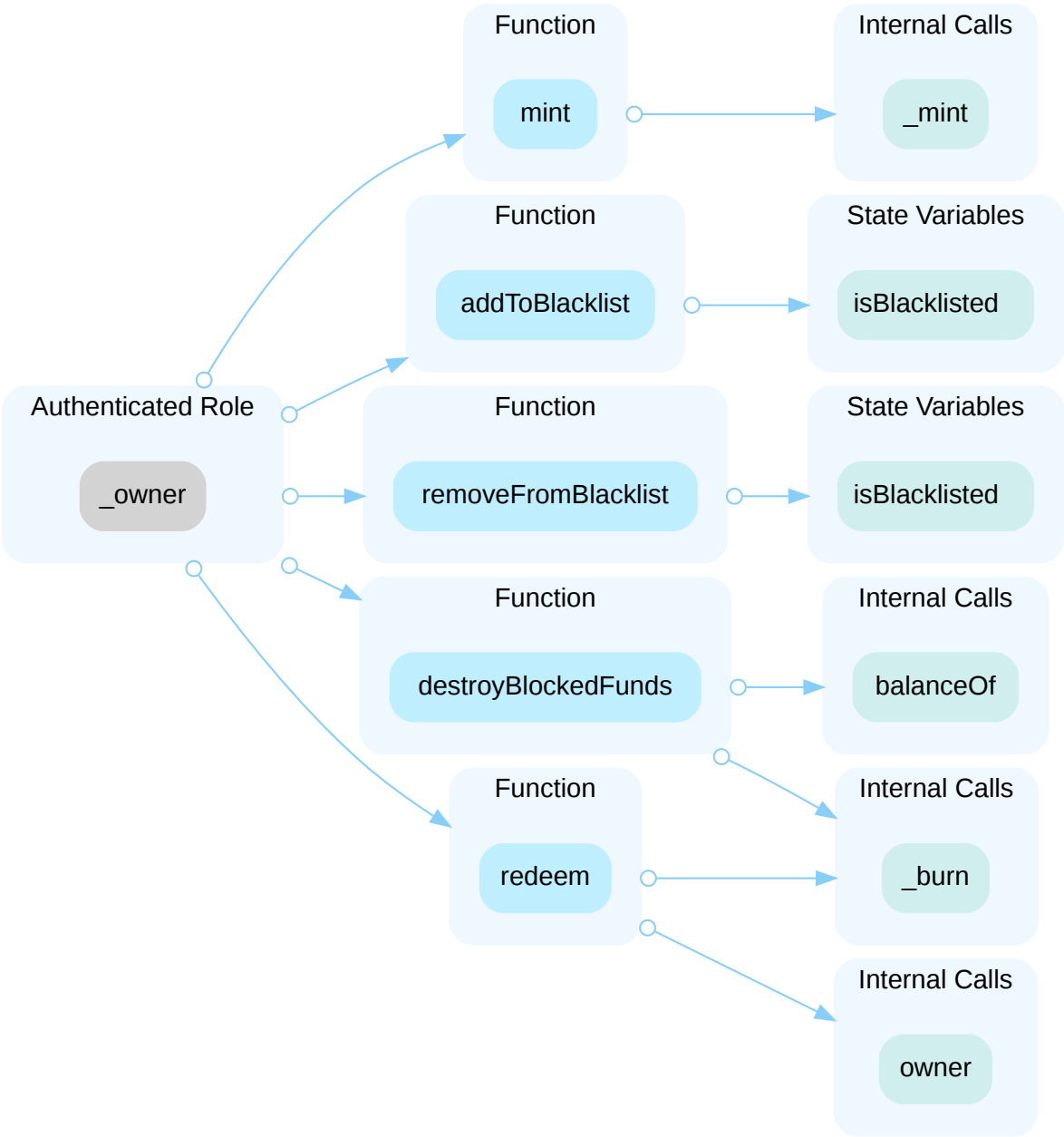
DDH-04 | CENTRALIZATION RELATED RISKS

Category	Severity	Location	Status
Centralization	**Major**	**contracts/DimentDollar.sol (DimentDollar): 73, 95, 119, 151, 164, 174, 189, 201**	**Mitigated**

Description

In the contract `DimentDollar` the role `_owner` has authority over the functions shown in the diagram below. Any compromise to the `_owner` account may allow the hacker to take advantage of this authority and :

- add or remove any address from the blacklist;
- mint any amount of token to any non-blacklisted address;
- burn all tokens from a blacklisted address;



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

[Diment team, 2024/02/28]: Issue acknowledged. I won't make any changes to the current version. We are a centralized token so we need these functions in our contract.

We will give ownership to a Multi-Signature Wallet contract whose implementation is included in the scope of this audit so hack risk is going minimum for mint and burn.

[CertiK, 2024/03/27]: 2024/03/27, 03:01:34 UTC, block 37323868 on Binance smart chain:

`DimentDollar` contract address is :

- **implementation**: 0xc545eed89bb404abbe5cfbf7c96643bcf4561309;
- **proxy**: 0x71b3a0566f4bf80331d115d8026a7022bf670cce;

The ownership of the **proxy** has been transferred, in this *transaction* to the `DimentTimelockController` deployed at address: 0xCfA0E2641Ce128959EbCDF680073E0C480B98442:

- its `DEFAULT_ADMIN_ROLE` is granted to addresses:
  - `DimentTimelockController`;
  - `DimentMultiSignatureWallet`;
- its `PROPOSER_ROLE` is granted to `DimentMultiSignatureWallet`;
- its `CANCELER_ROLE` is granted to `DimentMultiSignatureWallet`;
- its `EXECUTOR_ROLE` is granted to the signers of the `DimentMultiSignatureWallet`:
  1. 0xEb098A67D7c46cA48c701cd09d6A3A37b1BA0717 an EOA;
  2. 0x5D3C96bF7eCf9bDB75F18BEF5f4a7AEF351543Ea an EOA;
  3. 0xD5aE52e39750c52c94A725D7b7f717239d964AF5 an EOA;

the minimum delay is set to 48 hours.

The `DimentMultiSignatureWallet` has been deployed in this *transaction* with the following addresses as `owners_`:

1. 0xEb098A67D7c46cA48c701cd09d6A3A37b1BA0717 an EOA;
2. 0x5D3C96bF7eCf9bDB75F18BEF5f4a7AEF351543Ea an EOA;
3. 0xD5aE52e39750c52c94A725D7b7f717239d964AF5 an EOA;

`numConfirmationsRequired` is equal to 2.

The Diment team has applied the timelock with 48 hours, and the ⅔ multisig as a short-term solution. While this strategy has indeed reduced the risk, it's crucial to note that it has not completely eliminated it.

GOE-02 | CENTRALIZED CONTROL OF CONTRACT UPGRADE

Category	Severity	Location	Status
Centralization	Major	contracts/DimentDollar.sol (DimentDollar): 8; @openzeppelin/contracts/proxy/transparent/TransparentUpgradeableProxy.sol (TransparentUpgradeableProxy): 78~79, 94~98	Mitigated

Description

The contract `DimentDollar`, is used as the implementation logic used in an upgradable contract using the EIP-1967 Transparent Proxy pattern.

The `ProxyAdmin` has the authority to update the implementation contract and therefore change the logic of `DimentDollar`.

Any compromise to the `ProxyAdmin` account may allow a hacker to take advantage of this authority and change the implementation contract which is pointed by proxy and therefore execute potential malicious functionality in the implementation contract.

Recommendation

We recommend that the team make efforts to restrict access to the admin of the proxy contract. A strategy of combining a time-lock and a multi-signature (⅔, ⅗) wallet can be used to prevent a single point of failure due to a private key compromise. In addition, the team should be transparent and notify the community in advance whenever they plan to migrate to a new implementation contract.

Here are some feasible short-term and long-term suggestions that would mitigate the potential risk to a different level and suggestions that would permanently fully resolve the risk.

Short Term:

A combination of a time-lock and a multi signature (⅔, ⅗) wallet mitigate the risk by delaying the sensitive operation and avoiding a single point of key management failure.

- A time-lock with reasonable latency, such as 48 hours, for awareness of privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to a private key compromised;
  AND
- A medium/blog link for sharing the time-lock contract and multi-signers addresses information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the deployed time-lock address.
- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.
- Provide a link to the **medium/blog** with all of the above information included.

Long Term:

A combination of a time-lock on the contract upgrade operation and a DAO for controlling the upgrade operation mitigate the contract upgrade risk by applying transparency and decentralization.

- A time-lock with reasonable latency, such as 48 hours, for community awareness of privileged operations;
  AND
- Introduction of a DAO, governance, or voting module to increase decentralization, transparency, and user involvement;
  AND
- A medium/blog link for sharing the time-lock contract, multi-signers addresses, and DAO information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the deployed time-lock address.
- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.
- Provide a link to the **medium/blog** with all of the above information included.

Permanent:

Renouncing ownership of the `admin` account or removing the upgrade functionality can *fully* resolve the risk.

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

Note: we recommend the project team consider the long-term solution or the permanent solution. The project team shall make a decision based on the current state of their project, timeline, and project resources.

Alleviation

[Diment team, 2024/02/28]: Issue acknowledged. I won't make any changes to the current version. We are a centralized token so we need these functions in our contract.

We will give ownership to a Multi-Signature Wallet contract whose implementation is included in the scope of this audit so hack risk is going minimum for mint and burn.

[CertiK, 2024/03/27]: 2024/03/27, 03:01:34 UTC, block 37323868 on Binance smart chain:

The proxy address is 0x71b3a0566f4bf80331d115d8026a7022bf670cce, it has been deployed in the following *transaction*.

The **proxy admin** has been set to address 0xa72cA7922c30Db17d5aBd83F722569249BE5ef2D, whose owner is the EOA 0xE917A31C5941271834ba7Fff6d45bf223C8701d4.

[CertiK, 2024/03/27]: 2024/03/27, 17:03 UTC, block height 37340671 on Binance smart chain:

The team transferred ownership of the **proxy admin** to the **DimentTimelockController**: in this *transaction*.

The `DimentTimelockController` is deployed at the address: 0xCfA0E2641Ce128959EbCDF680073E0C480B98442:

- its `DEFAULT_ADMIN_ROLE` is granted to addresses:
  - `DimentTimelockController`;
  - `DimentMultiSignatureWallet`;
- its `PROPOSER_ROLE` is granted to `DimentMultiSignatureWallet`;
- its `CANCELER_ROLE` is granted to `DimentMultiSignatureWallet`;
- its `EXECUTOR_ROLE` is granted to the signers of the `DimentMultiSignatureWallet`:
  1. 0xEb098A67D7c46cA48c701cd09d6A3A37b1BA0717 an EOA;
  2. 0x5D3C96bF7eCf9bDB75F18BEF5f4a7AEF351543Ea an EOA;
  3. 0xD5aE52e39750c52c94A725D7b7f717239d964AF5 an EOA;

the minimum delay is set to 48 hours.

The `DimentMultiSignatureWallet` has been deployed in this *transaction* with the following addresses as `owners_`:

1. 0xEb098A67D7c46cA48c701cd09d6A3A37b1BA0717 an EOA;
2. 0x5D3C96bF7eCf9bDB75F18BEF5f4a7AEF351543Ea an EOA;
3. 0xD5aE52e39750c52c94A725D7b7f717239d964AF5 an EOA;

`numConfirmationsRequired` is equal to 2.

The Diment team has applied the timelock with 48 hours, and the ⅔ multisig as a short-term solution. While this strategy has indeed reduced the risk, it's crucial to note that it has not completely eliminated it.

DDH-02 | UNPROTECTED INITIALIZER

Category	Severity	Location	Status
Coding Issue	Minor	contracts/DimentDollar.sol (DimentDollar): 36	Resolved

Description

One or more logic contracts do not protect their initializers. An attacker can call the initializer and assume ownership of the logic contract, whereby she can perform privileged operations that trick unsuspecting users into believing that she is the owner of the upgradeable contract.

```
8  contract DimentDollar is
```

- `DimentDollar` is an upgradeable contract that does not protect its initializer.

```
36     function initialize(
```

- `initialize` is an unprotected initializer function.

Recommendation

We advise calling `_disableInitializers` in the constructor or giving the constructor the `initializer` modifier to prevent the initializer from being called on the logic contract.

Reference: https://docs.openzeppelin.com/upgrades-plugins/1.x/writing-upgradeable#initializing_the_implementation_contract

Alleviation

[CertiK, 2024/03/01]: The team heeded the advice and resolved the issue in commit 1d1527209092cdd04b3576a4a5f583281a57589b.

OPTIMIZATIONS | DIMENT - (MAIN CONTRACT & PROXY)

ID	Title	Category	Severity	Status
DDH-03	Inefficient Storage Access In Loops	Gas Optimization	Optimization	Resolved

DDH-03 | INEFFICIENT STORAGE ACCESS IN LOOPS

Category	Severity	Location	Status
Gas Optimization	Optimization	contracts/DimentDollar.sol (DimentDollar): 123, 126, 126, 127, 133, 140	Resolved

Description

The current implementation frequently accesses dynamic array lengths within loop conditions, leading to redundant computation and potential gas inefficiencies. Each iteration of the loop retrieves the array's length from storage, incurring unnecessary gas costs, especially in loops executed.

Recommendation

We recommend employing length caching, by storing the array's length in a local variable before the loop starts and using this variable in the loop condition this will optimize gas cost.

Alleviation

[CertiK, 2024/03/16]: The team resolved the finding in commit: 594bb5d5197914fbd8655f272154050c3d73be6c.

APPENDIX | DIMENT - (MAIN CONTRACT & PROXY)

Finding Categories

Categories	Description
Gas Optimization	Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.
Coding Issue	Coding Issue findings are about general code quality including, but not limited to, coding mistakes, compile errors, and performance issues.
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.