CERTIK

Preliminary Comments

Xwin Finance

Apr 13th, 2022

Table of Contents

Summary

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

- 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.

Overview

Project Summary

Project Name	Xwin Finance
Platform	BSC
Language	Solidity
Codebase	https://github.com/xwinfinance/ToppyMarketplace
Commit	

Audit Summary

Delivery Date	Apr 13, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

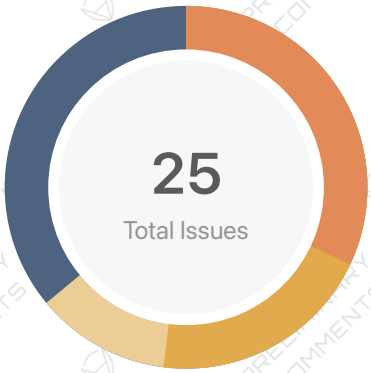
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Mitigated	Partially Resolved	Resolved
Critical	0	0	0	0	0	0	0
Major	8	0	0	6	0	0	2
Medium	5	1	0	1	0	0	3
Minor	3	0	0	1	0	0	2
Informational	9	0	0	2	0	0	7
Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
IBE	ToppyMarketplace/contracts/IBEP20.sol	23b464018f5fe38c7ef8c98e2211be384a2e6dec1502cfc3bf27e9ec0b7575e2
TST	ToppyMarketplace/contracts/ToppyStaking.sol	7d7bf7065ef17a976cf3309d64a19d6442940b2e5d3eb5e7c0fa2e6d995951b3
BEP	ToppyMarketplace/contracts/BEP20.sol	0265ed53686ad2762d555381c11082b1c54fab76805b33eeb9d3b99da734c1d3
TMT	ToppyMarketplace/contracts/ToppyMarketplace.sol	c1b53d86235a986bcabbadbcc876150a433049bbc057eb71bbe503d764593fca
TMS	ToppyMarketplace/contracts/ToppyMasterSetting.sol	cb06e142e059a852e02ac7ca0af4e1c6cdc73e4ad22e362d062506120256a540
TSP	ToppyMarketplace/contracts/ToppySupportPayment.sol	a5043cfa348f3979a9f8936a446d149885d48f5489e08851fd05027f5bdc9b6a
THT	ToppyMarketplace/contracts/TransferHelper.sol	d9b8824624ec34b2fe7dfc7db9c2c38d08d5dc3a44c0a794757fecf4844ea739
TSN	ToppyMarketplace/contracts/ToppyStandardNFT.sol	66fd698f7d97f53e379b5e5141bdb186685cc2ec6401a7dcb509658e74ec3802
ERC	ToppyMarketplace/contracts/ERC721.sol	a2669d682ee6a3e41ea7a7fb5e921c9782f8e01bc1720a63df5c1250e7fb9a2f
TMC	ToppyMarketplace/contracts/tests	
TMP	ToppyMarketplace/contracts/tests/ToppyMarketPlace_test.sol	bb7483e82b20cef7e20e70de071661c681bc68e2dc7669aa47197117a292d840
ITM	ToppyMarketplace/contracts/IToppyMint.sol	d9b422184d1482a7ba50e418574377036e50bb21a23d812ceef878bb8260f8da
TMN	ToppyMarketplace/contracts/ToppyMysteriousNFT.sol	2e97e881495b0a81b745d04fdd3d75a0f6fe2d3bc63a00fab9f5ba8b9f65711e
TMM	ToppyMarketplace/contracts/ToppyMint.sol	01930bc65d4d5f228210eeb209a1e7fa3dd7581e321a2c9822edef3eab0fd52d

Findings



Critical	0 (0.00%)
Major	8 (32.00%)
Medium	5 (20.00%)
Minor	3 (12.00%)
Informational	9 (36.00%)
Discussion	0 (0.00%)

ID	Title	Category	Severity	Status
TMM-01	Change The Way To Calculate The `managerFee`	Logical Issue	Informational	Resolved
TMN-01	Centralization Risk In ToppyMysteriousNFT.sol	**Centralization / Privilege**	**Major**	Acknowledged
TMN-02	Update `maxSupply` Is Unsafe	Logical Issue	Medium	Resolved
TMN-03	Lack Of Input Validation	Volatile Code	Informational	Resolved
TMS-01	Centralization Risk In ToppyMasterSetting.sol	**Centralization / Privilege**	**Major**	Acknowledged
TMS-02	`platformComm` Fee Not Capped	Logical Issue	Informational	Resolved
TMS-03	Lack Of Input Validation	Logical Issue	Informational	Acknowledged
TMT-01	`offer` Should Only Be Called For `ListingType.English` Listing	Logical Issue	Major	Resolved
TMT-02	`bid` Should Not Be Called For `ListingType.English` Keys	Logical Issue	Major	Resolved
TMT-03	Centralization Risk In ToppyMarketplace.sol	**Centralization / Privilege**	**Major**	Acknowledged
TMT-04	The `listing_` With Offers Could Not Be Canceled	Logical Issue	Medium	Resolved

TMT-05	Owner Should Not Cancel The Listing With Offers	Logical Issue	Medium	Resolved
TMT-06	Lack Of Repeated Listing Validation	Logical Issue	Minor	Resolved
TMT-07	The `highestOff.buyer` Also Can Call `acceptOffer`	Logical Issue	Minor	Resolved
TMT-08	Why `onlyAdminExecutor` Can Call `cancelListingByAdmin`	Logical Issue	Minor	Acknowledged
TMT-09	Use `type(uint128).max` Is More Readable	Language Specific	Informational	Resolved
TMT-10	`_listingParams.tokenPayment` Should Be In `supportPayment.isEligibleToken`	Logical Issue	Informational	Resolved
TMT-11	Make Sure `highestOff` Is Valid	Logical Issue	Informational	Resolved
TSN-01	Centralization Risk In ToppyStandardNFT.sol	**Centralization / Privilege**	**Major**	Acknowledged
TSN-02	Check Effect Interaction Pattern Violated	Logical Issue	Medium	Pending
TSP-01	Centralization Risk In ToppySupportPayment.sol	**Centralization / Privilege**	**Major**	Acknowledged
TST-01	Centralization Risk In ToppyStaking.sol	**Centralization / Privilege**	**Major**	Acknowledged
TST-02	No New Tokens Minted	Logical Issue	Medium	Acknowledged
TST-03	No Need To Use Library `SafeMath`	Language Specific	Informational	Acknowledged
TST-04	`user.amount` Is Always Zero	Logical Issue	Informational	Resolved

TMM-01 | Change The Way To Calculate The `managerFee`

Category	Severity	Location	Status
Logical Issue	Informational	ToppyMarketplace/contracts/ToppyMint.sol: 116	Resolved

Description

In the function `_payFee`, `managerFee = totalAmount * nft.managerComm() / 10000;` may not include all remain tokens.

Recommendation

We recommend using subtraction to calculate `managerFee`.

```
1    uint managerFee = totalAmount - creatorFee - platformFee;
```

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

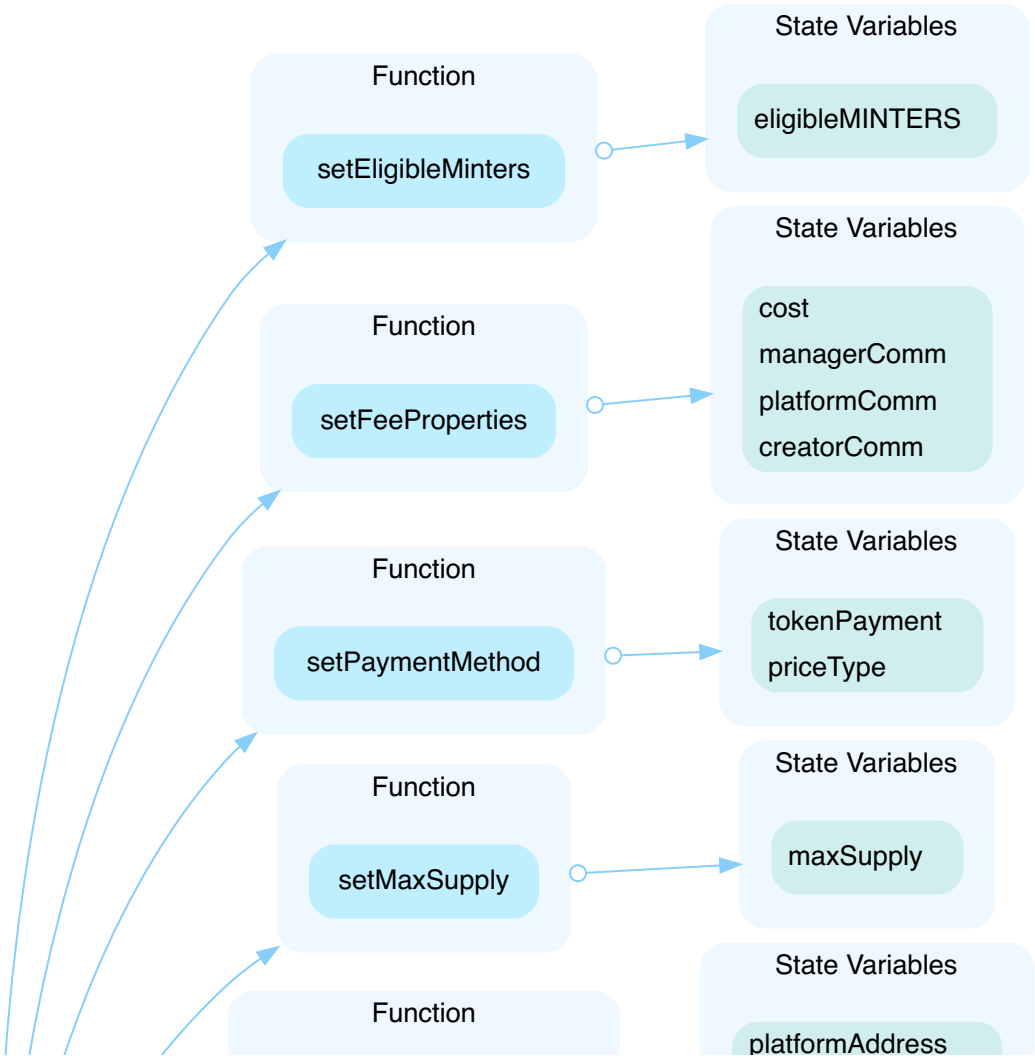
TMN-01 | Centralization Risk In ToppyMysteriousNFT.sol

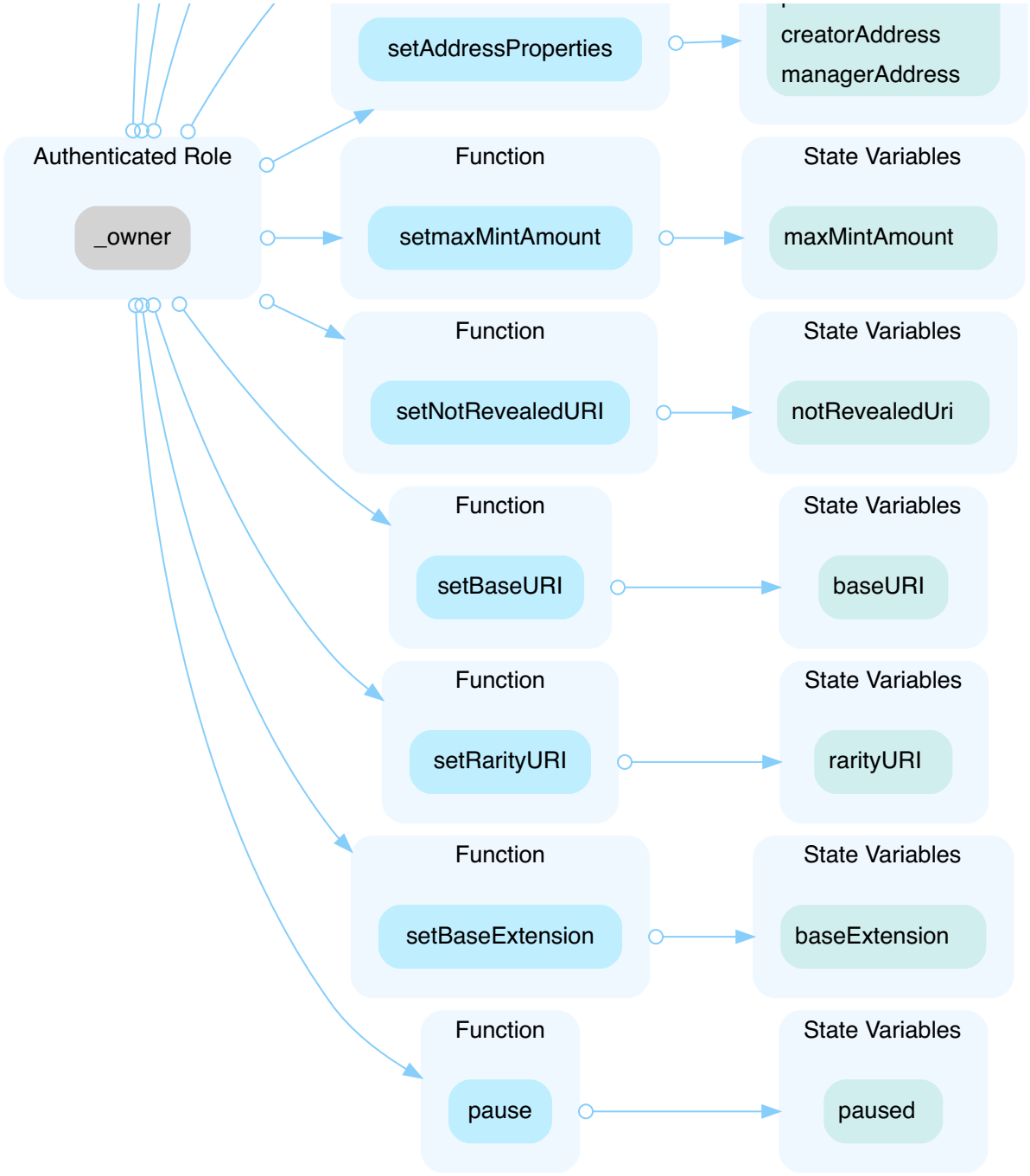
Category	Severity	Location	Status
Centralization / Privilege	Major	ToppyMarketplace/contracts/ToppyMysteriousNFT.sol: 92~94, 181~186, 188~191, 193~195, 197~201, 203~205, 207~209, 211~213, 215~217, 219~221, 223~225	Acknowledged

Description

In the contract `ToppyMysteriousNFT` 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.





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

[XWIN Developer] Once live, the Mystery NFT owner will be revoked. No further changes can be done once it is all sold out.

TMN-02 | Update `maxSupply` Is Unsafe

Category	Severity	Location	Status
Logical Issue	Medium	ToppyMarketplace/contracts/ToppyMysteriousNFT.sol: 194	Resolved

Description

The function `setMaxSupply` can update `maxSupply` to any value, `_maxSupply` maybe less than `totalSupply()`, even less than the minted tokenId.

Recommendation

We recommend adding the validation.

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

TMN-03 | Lack Of Input Validation

Category	Severity	Location	Status
Volatile Code	Informational	ToppyMarketplace/contracts/ToppyMysteriousNFT.sol: 182~185	Resolved

Description

The input parameter should be validated according to the business logic, like `managerComm + platformComm + creatorCom` should be 10000 if the owner makes a mistake.

Recommendation

We recommend adding input validation.

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

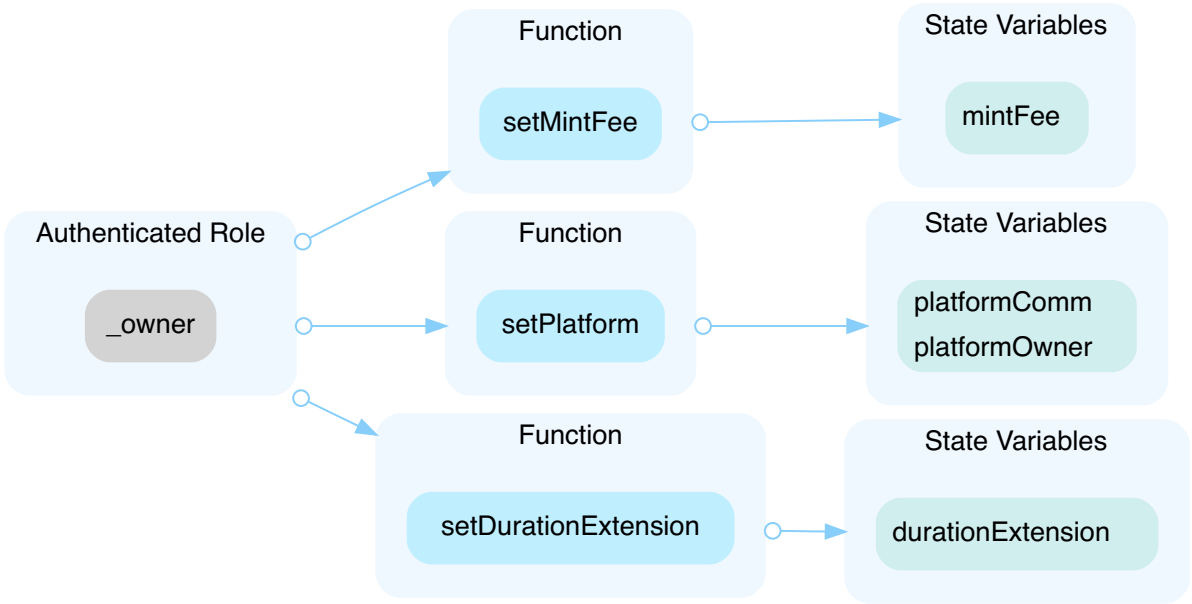
TMS-01 | Centralization Risk In ToppyMasterSetting.sol

Category	Severity	Location	Status
Centralization / Privilege	Major	ToppyMarketplace/contracts/ToppyMasterSetting.sol: 20~22, 23~26, 27~29	Acknowledged

Description

In the contract `ToppyMaster` 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.



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

[XWIN Developer] The master setting is for the admin to update mint or platform fees as strategies change in the long run. We would plan to have the mint fee set to zero or set back to some number based on the business requirement. Anyhow, the platform fee is capped at 10% as suggested. The team is currently using a cold wallet, and it will transfer to the multi-sign wallet shortly.

TMS-02 | `platformComm` Fee Not Capped

Category	Severity	Location	Status
Logical Issue	Informational	ToppyMarketplace/contracts/ToppyMasterSetting.sol: 24	Resolved

Description

The owner can update `platformComm` to any value greater than the base value(10000).

Recommendation

We recommend adding an upper limit.

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

TMS-03 | Lack Of Input Validation

Category	Severity	Location	Status
Logical Issue	Informational	ToppyMarketplace/contracts/ToppyMasterSetting.sol: 28	Acknowledged

Description

The owner can call `setDurationExtension` to update `durationExtension`, the input parameter `_durationExtension` can be any value including zero. If `durationExtension` is zero, it may affect the process of `English Auction`.

The fix commit added the upper limit, lower limit is also needed.

```
1    require(_durationExtension <= 86400, "max duration extension 24hours");
```

Recommendation

We recommend adding a low limit validation.

TMT-01 | `offer` Should Only Be Called For `ListingType.English` Listing

Category	Severity	Location	Status
Logical Issue	Major	ToppyMarketplace/contracts/ToppyMarketplace.sol: 388	Resolved

Description

The function `offer` can be called for keys with any `listingType`, but according to the logic, it can only be called with `ListingType.English` keys.

Recommendation

We recommend adding `listingType` validation.

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

TMT-02 | `bid` Should Not Be Called For `ListingType.English` Keys

Category	Severity	Location	Status
Logical Issue	Major	ToppyMarketplace/contracts/ToppyMarketplace.sol: 433	Resolved

Description

If the `bid` function is called for `ListingType.English` keys, the user can bypass the auction process and take the NFT token at a low price.

Recommendation

We recommend adding `listingType` validation.

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

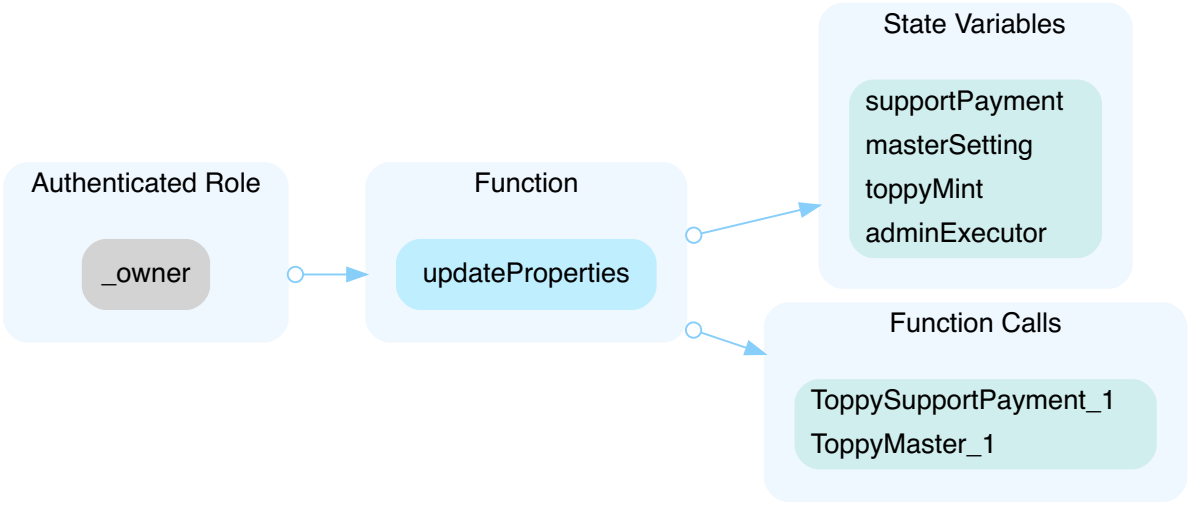
TMT-03 | Centralization Risk In ToppyMarketplace.sol

Category	Severity	Location	Status
Centralization / Privilege	Major	ToppyMarketplace/contracts/ToppyMarketplace.sol: 98~108, 323~328, 356~360	Acknowledged

Description

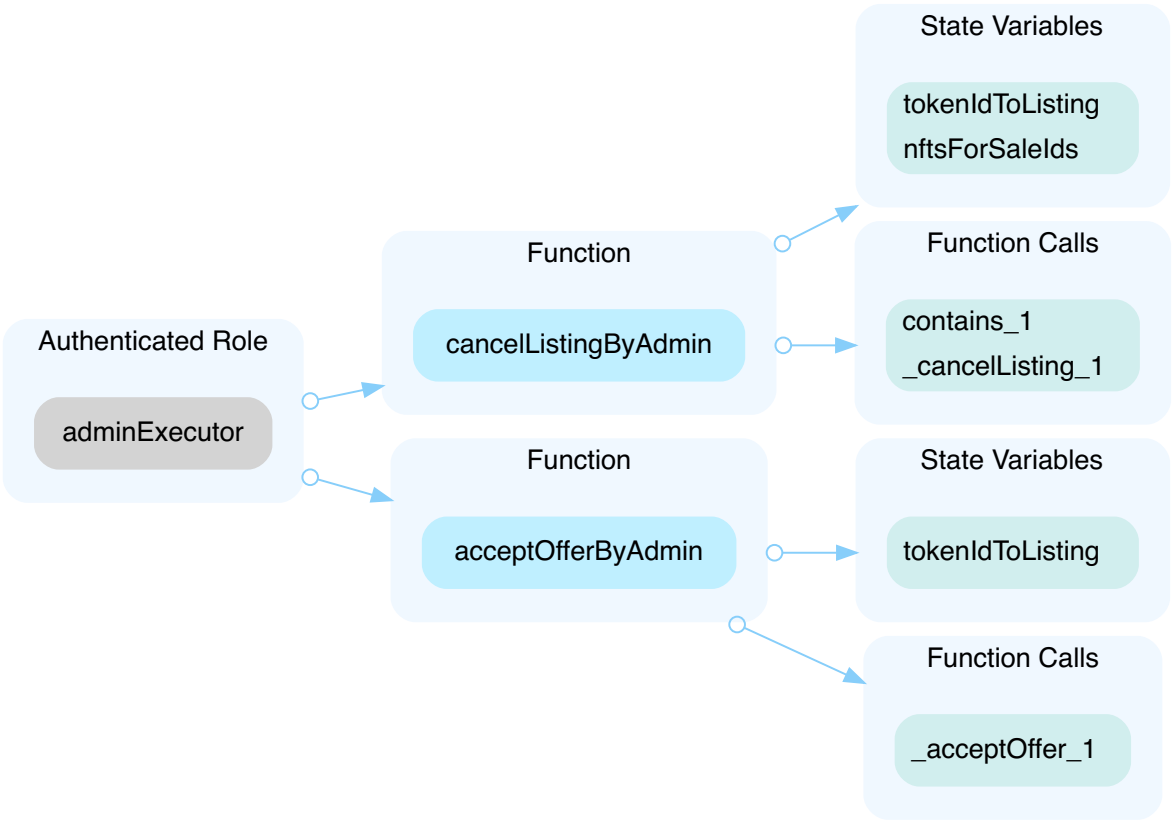
In the contract `ToppyMarketPlace` the role `_owner` has authority over the functions shown in the diagram below.

Any compromise to the `_owner` account may allow the hacker to take advantage of this authority.



In the contract `ToppyMarketPlace` the role `adminExecutor` has authority over the functions shown in the diagram below.

Any compromise to the `adminExecutor` account may allow the hacker to take advantage of this authority.



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.

TMT-04 | The `listing_` With Offers Could Not Be Canceled

Category	Severity	Location	Status
Logical Issue	Medium	ToppyMarketplace/contracts/ToppyMarketplace.sol: 336	Resolved

Description

The `cancelListingByKey` function can cancel `listing_`, which may have offer in the `highestOffer[listing_.key]`. It is unreasonable, and it also makes `highestOffer[listing_.key].buyer` can not withdraw their tokens.

Recommendation

We recommend adding a limit to make sure there is no offer in the `highestOffer[listing_.key]`.

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

TMT-05 | Owner Should Not Cancel The Listing With Offers

Category	Severity	Location	Status
Logical Issue	Medium	ToppyMarketplace/contracts/ToppyMarketplace.sol: 350~352	Resolved

Description

The owner can call the function `cancelListingByKey` to cancel any `listing_` with any state, even the `listing_` has the `highestOffer` and the auction process is over. It seems not reasonable from the point of fairness.

Recommendation

We advise that the `listing_` with `highestOffer` should not be canceled.

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

TMT-06 | Lack Of Repeated Listing Validation

Category	Severity	Location	Status
Logical Issue	Minor	ToppyMarketplace/contracts/ToppyMarketplace.sol: 195	Resolved

Description

Users can call `_createListing` with the same `_listingParams.tokenId`, which will update an existing `listing`. Updating a `listing` that is in the auction process is unreasonable.

Recommendation

We recommend fixing this according to the business logic.

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

TMT-07 | The `highestOff.buyer` Also Can Call `acceptOffer`

Category	Severity	Location	Status
Logical Issue	Minor	ToppyMarketplace/contracts/ToppyMarketplace.sol: 365	Resolved

Description

The function `acceptOffer` can only be called by the `listing_.tokenId` owner, the `highestOff.buyer` should also have the right to call it.

Recommendation

We recommend adding `highestOff.buyer`.

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

TMT-08 | Why `onlyAdminExecutor` Can Call `cancelListingByAdmin`

Category	Severity	Location	Status
Logical Issue	Minor	ToppyMarketplace/contracts/ToppyMarketplace.sol: 323	Acknowledged

Description

Why admin executor can cancel user's `listing_`?

Recommendation

We recommend removing these functions.

TMT-09 | Use `type(uint128).max` Is More Readable

Category	Severity	Location	Status
Language Specific	Informational	ToppyMarketplace/contracts/ToppyMarketplace.sol: 168~171, 183~185	Resolved

Description

For an integer type X, you can use type(X).min and type(X).max to access the minimum and maximum value representable by the type.

Recommendation

We recommend using language specified maximum instead.

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

TMT-10 | `_listingParams.tokenPayment` Should Be In `supportPayment.isEligibleToken`

Category	Severity	Location	Status
Logical Issue	Informational	ToppyMarketplace/contracts/ToppyMarketplace.sol: 261	Resolved

Description

According to the logic of `createListing`, `_listingParams.tokenPayment` should be in `supportPayment.isEligibleToken` when calling `updateBulkListing`.

Recommendation

We recommend addint this validation.

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

TMT-11 | Make Sure `highestOff` Is Valid

Category	Severity	Location	Status
Logical Issue	Informational	ToppyMarketplace/contracts/ToppyMarketplace.sol: 374	Resolved

Description

In the function `_acceptOffer`, there is lack of an explicit validation for `highestOff`, it maybe none.

Recommendation

We recommend adding an explicit validation for `highestOff`.

Alleviation

Fixed in commit `fff34be64f4d4d80411a36789791c9bb8e05cb44`.

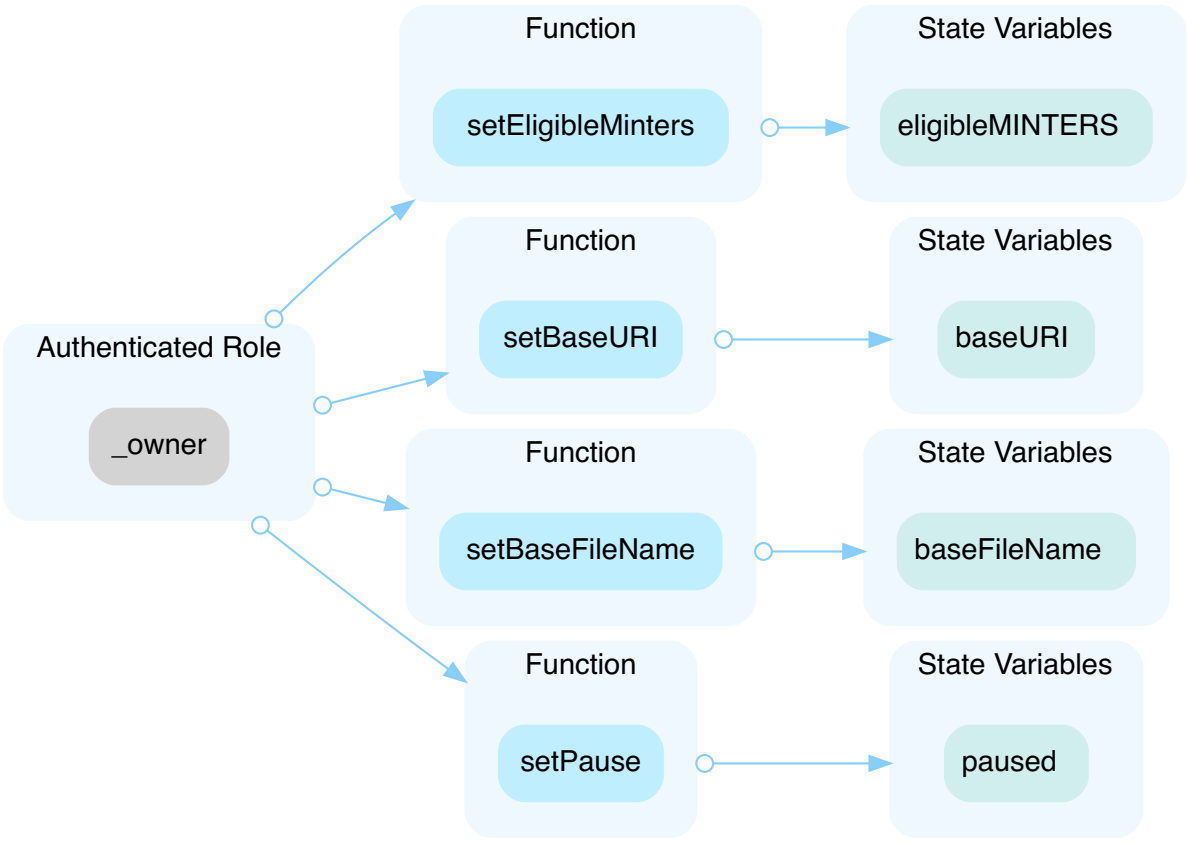
TSN-01 | Centralization Risk In ToppyStandardNFT.sol

Category	Severity	Location	Status
Centralization / Privilege	Major	ToppyMarketplace/contracts/ToppyStandardNFT.sol: 30~32, 83~85, 87~89, 91~93	Acknowledged

Description

In the contract `ToppyStandardNFT` 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.



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

[XWIN Developer] It is needed to configure the handshake between smart contracts initially. The

owner will be revoked permanently once the NFT is live.

TSN-02 | Check Effect Interaction Pattern Violated

Category	Severity	Location	Status
Logical Issue	Medium	ToppyMarketplace/contracts/ToppyStandardNFT.sol: 49	Pending

Description

The order of external call and storage manipulation must follow the check-effect-interaction pattern.

Recommendation

We advise the client to check if storage manipulation is before the external call/transfer operation.LINK

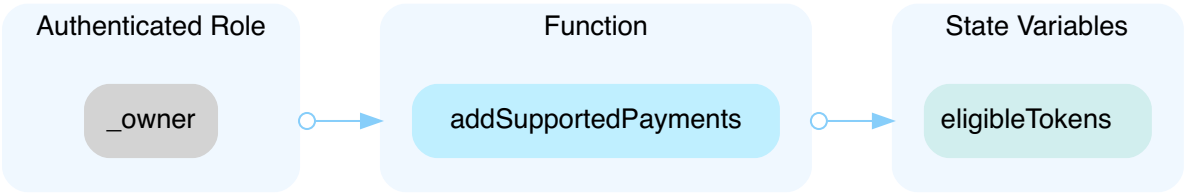
TSP-01 | Centralization Risk In ToppySupportPayment.sol

Category	Severity	Location	Status
Centralization / Privilege	Major	ToppyMarketplace/contracts/ToppySupportPayment.sol: 16~21	Acknowledged

Description

In the contract `ToppySupportPayment` 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.



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

[XWIN Developer] Based on the nature of the business, the admin needs to reserve the right to add a new support payment token in the future. Therefore, the admin needs to be able to add new tokens as new payments. The current team wallet is a hardware cold wallet and is being transferred to a multi-signature wallet for the long term.

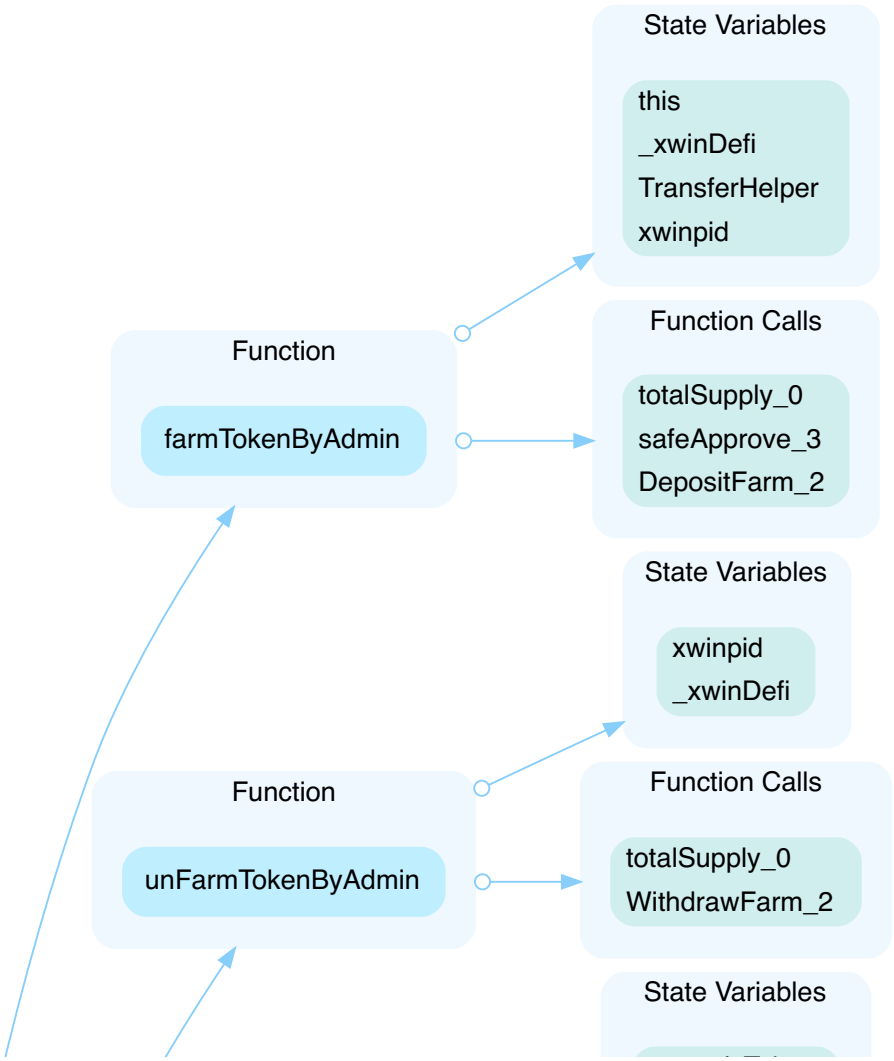
TST-01 | Centralization Risk In ToppyStaking.sol

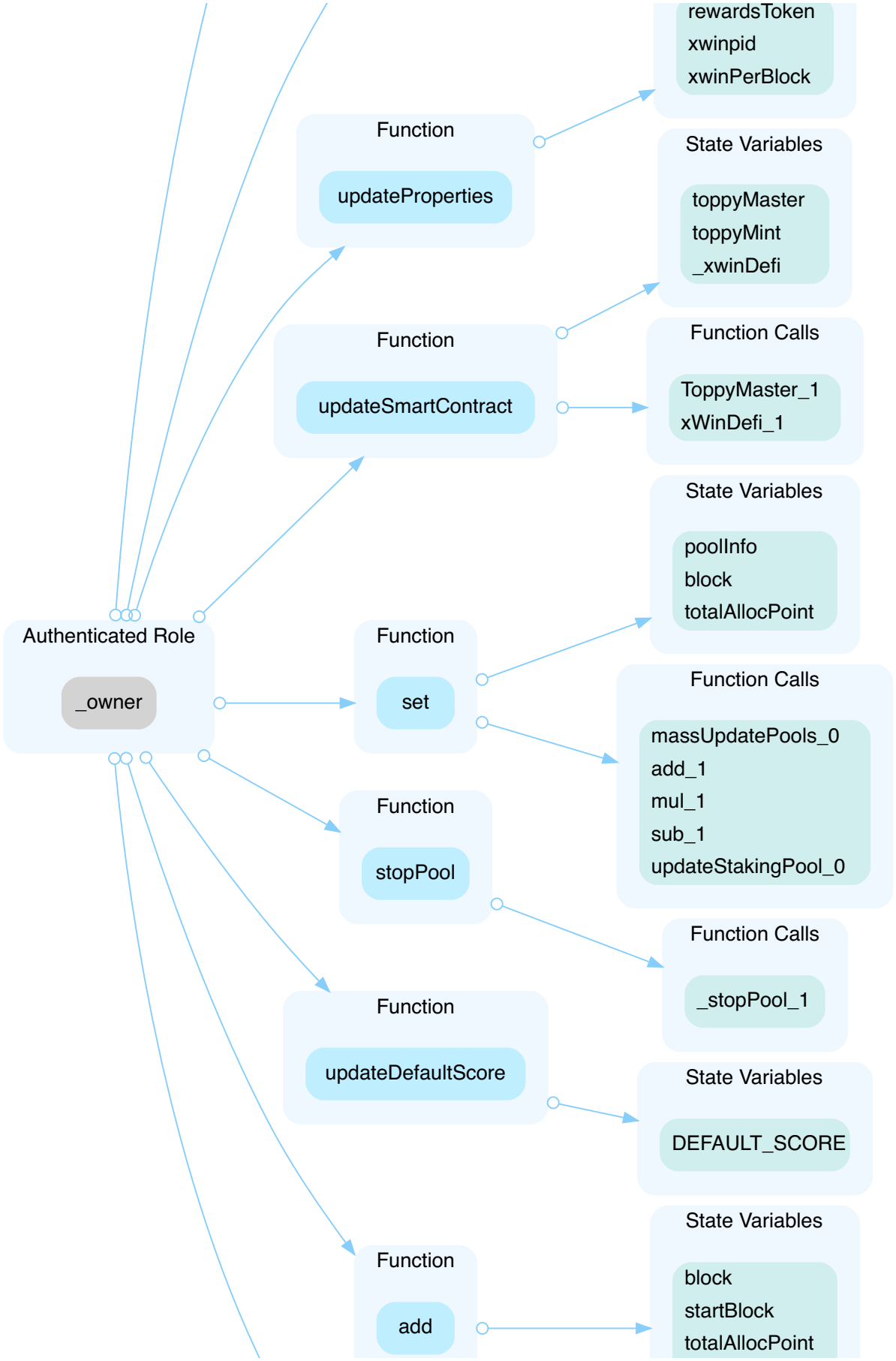
Category	Severity	Location	Status
Centralization / Privilege	Major	ToppyMarketplace/contracts/ToppyStaking.sol: 106~109, 111~113, 116~125, 131~136, 139~150, 153~155, 157~159, 172~187, 323~334	Acknowledged

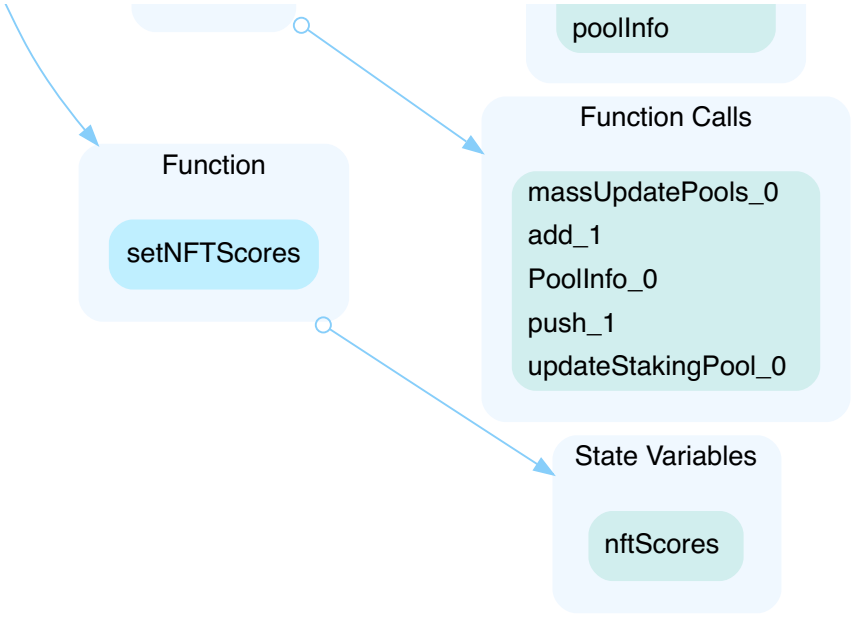
Description

In the contract `ToppyStaking` 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.




rewardsToken
xwinpid
xwinPerBlock
Function
updateProperties
State Variables
toppyMaster
toppyMint
_xwinDefi
Function
updateSmartContract
Function Calls
ToppyMaster_1
xWinDefi_1
State Variables
poolInfo
block
totalAllocPoint
Authenticated Role
_owner
Function
set
Function Calls
massUpdatePools_0
add_1
mul_1
sub_1
updateStakingPool_0
Function
stopPool
Function Calls
_stopPool_1
Function
updateDefaultScore
State Variables
DEFAULT_SCORE
Function
add
State Variables
block
startBlock
totalAllocPoint



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

[XWIN Developer] ToppyStaking is the protocol that can collaborate with different projects for their NFT to be staked to earn XWIN token. Due to the business nature needs, the admin needs to add new pools or set the NFT rarity score. So, we need the ability to update it as part of the business model.

TST-02 | No New Tokens Minted

Category	Severity	Location	Status
Logical Issue	Medium	ToppyMarketplace/contracts/ToppyStaking.sol: 103~113	Acknowledged

Description

Only one token is minted in the contstructor, no new tokens minted in the contract. The function `farmTokenByAdmin` can only deposit one token to `_xwinDefi`. `totalSupply` is used in `farmTokenByAdmin` and `unFarmTokenByAdmin` functions is always `1*10**18`.

Recommendation

We recommend fixing this issue.

Alleviation

[XWIN developer] Based on the logic of integrating the existing XWIN protocol and Toppy Staking. ToppyStaking is the subset of the contract that will get the token emission from xWIN defi protocol. So, there needs to have a handshake between both of the protocols. Once set up, ToppyStaking can harvest the XWIN token from xWIN Defi protocol. This is by design.

TST-03 | No Need To Use Library `SafeMath`

Category	Severity	Location	Status
Language Specific	Informational	ToppyMarketplace/contracts/ToppyStaking.sol: 32	Acknowledged

Description

Solidity v0.8.0 and later versions check underflow/overflow by default, and therefore the library `SafeMath` is not necessary.

Source: link

Recommendation

We recommend using the default arithmetic check instead of the library `SafeMath`.

TST-04 | `user.amount` Is Always Zero

Category	Severity	Location	Status
Logical Issue	Informational	ToppyMarketplace/contracts/ToppyStaking.sol: 289~294	Resolved

Description

The `_stake` function executes successfully only if `user.amount == 0`, so the aforementioned code will never be executed.

Recommendation

We recommend removing unused code.

Alleviation

Fixed in commit ``.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

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

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.

About

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.